

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-005614

(43)Date of publication of application : 12.01.2001

(51)Int.Cl.

G06F 3/06

G06F 12/00

G06F 12/14

(21)Application number : 11-179620

(71)Applicant : HITACHI LTD

(22)Date of filing : 25.06.1999

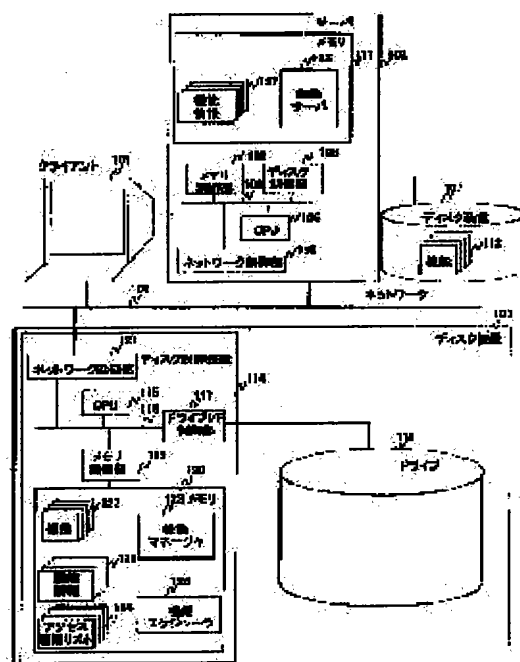
(72)Inventor : WATANABE NAOKI
TAKAMOTO YOSHIFUMI
ODAWARA HIROAKI

(54) DISK DEVICE AND SERVER DEVICE

(57)Abstract:

PROBLEM TO BE SOLVED: To make effectively restrictable a data access by informing a disk device of access limitation at the time of outputting a function execution request to the disk device.

SOLUTION: A client 101 transmits a function to be executed to a server 102 and simultaneously transmits also information on such as a disk device 103 to be a transmission destination, a function name and a version. The server 102 stores the received function, prepares function information 127 by using the received information, prepares function information 126 having an access range list 124 to be sent to the display device 103, and transmits the function and the function information 126 to the device 103. The device 103 stores these pieces of information. If a parameter also is simultaneously sent when a function execution request is sent from the client 101 to the server 102, the server 102 further prepares the function information 126, outputs the function execution request to the device 103 and sends the function information 126. The device 103 executes the function and executes access limitation at the execution of the function in accordance with the function information 126.



* NOTICES *

JP0 and INPIT are not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.*** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

CLAIMS

[Claim(s)]

[Claim 1]A disk unit connected to a server and a client apparatus via a network, comprising:

A storage for memorizing data.

A means by which it has a control device and this control device controls input and output to said storage.

A means to store in a memory function data about a function which it was transmitted from said server and received, and this function.

A means to perform this function according to an execution demand of said function from said server, and a limit means which restricts access to data memorized by said storage at the time of this functional execution based on said function data.

[Claim 2]A disk unit, wherein said function data have a list in which an accessible range is shown in the disk unit according to claim 1 and said limit means restricts access based on this list.

[Claim 3]A disk unit, wherein each item of said list has a lead, a light, and an attribute about access restriction at the time of functional execution of ** which can be performed in the disk unit according to claim 2.

[Claim 4]A disk unit, wherein said control device has a means to which abnormal termination of the execution of said function is carried out when access which breaks at restriction of said access occurs in the disk unit according to claim 2.

[Claim 5]A disk unit comprising of either claim 1 thru/or claim 4 given in a claim:

A means by which said control device supervises whether execution of said function was performed normally.

A means to return data memorized by said storage when execution of said function is not performed normally to a state before performing said function.

[Claim 6]A disk unit comprising of either claim 1 thru/or claim 4 given in a claim:

A means by which said control device supervises whether execution of said function was performed normally.

A means to set directions of a user of whether to return data memorized by said storage when execution of said function is not performed normally to a state before performing said function as said function data.

A means to return data memorized by said storage to a state before performing said function only when directions which return data memorized by said storage to a state before performing said function when execution of said function is not performed normally are set as these function data.

[Claim 7]A disk unit, wherein said control device does not overwrite update information based on execution of a function in the disk unit according to claim 5 or 6 at data before updating until execution of said function is completed.

[Claim 8]A disk unit, wherein said control device stores update information based on execution of said function in a memory in this control device in the disk unit according to claim 7.

[Claim 9]A server apparatus connected to a client apparatus and a disk unit via a network, comprising:

A means to generate function data which restrict an access range to data memorized by storage of said disk unit for every function execution request when an execution demand of a function in said disk unit is received from said client.

A means to transmit these function data to said disk unit.

[Claim 10]A means for this server apparatus to receive user ID information at least from said client apparatus, and to generate said function data in the server apparatus according to claim 9, A server apparatus generating received this function data which restrict said access range based on user ID information at least.

[Claim 11]A disk unit connected with a client apparatus via a network, comprising:
A storage for memorizing data.

A means by which it has a control device and this control device receives a function execution request and user ID information from a client apparatus via a network.

A means to generate function data which restrict an access range to data memorized by said storage for every function execution request based on this user ID information, and to restrict an access range based on these function data.

[Translation done.]

* NOTICES *

IPD and INPIT are not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.*** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]
[Field of the Invention]This invention relates to the disk unit which makes possible data protection when starting the disk unit which carried out direct continuation to the network, especially carrying out off-road one of the function to this disk unit and performing this function, and a related server apparatus.

[0002]
[Description of the Prior Art]Drawing 1 of the example of this invention is diverted and it explains below. By improvement in the high integration art of LSI, loading of a processor more highly efficient than before or highly efficient control LSI has been attained also at the disk unit 103. Then, network 104 I/F is provided in the disk unit 103, and not only the server 102 but the network 104 connection disk unit 103 which makes direct access possible also from the client 101, and raises the throughput of the whole system is proposed. There is NASD (Network-Attached Secure Disks) which Garth A. Gibson and others of CMU (Carnegie Mellon University) has proposed. ACM International Conference on Measurement and Modeling of Computer Systems (Sigmetrics'97), It is detailed to "File Server Scaling with Network-Attached Secure Disks" in Seattle, Washington, June 15-18, and 1997. Carry out off-road one of the processing which the server 102 is performing to the disk unit 103, reduce the loads of the server 102, and the intelligent disk unit 103 that a system throughput will be improved is proposed, There is Active Disks which Erik Riedel of CMU, Garth Gibson and others have proposed. Active Disks Theconference paper "Active Storage For Large-Scale Data Mining and Multimedia, It is detailed to "Proc. of the 24 th International Conference on Very large Databases (VLDB'98), New York, New York, August 24-27, and 1998.

[0003]
[Problem(s) to be Solved by the Invention]When carrying out off-road one of the function to the disk unit 103 and performing it to it, restrictions, such as execution authority at the time of execution and an access permission of data, are difficult. The selection/extracting processing to the inquiry from a client [in / here / with a function / a database], It is the processing which data conversion, such as transmission of the immediate data between the client disk units which do not go via a server, double-izing of the data between disk units, and ECU<=>JIS, etc. occur, and started the processing currently performed in the server and whose execution was enabled on the disk unit. Although management according to the existing remote procedure processing which is being used by RPC (Remote Procedure Call) which is used by Unix is employable, since OS, a vendor, etc. will be restricted, the correspondence to OS of a different kind is difficult. Under the present circumstances, although there is Unix RPC (Remote Procedure Call) with the nearest gestalt, There is the necessity of acquiring the access permission of a file to a server whenever it will open a file, if RPC is diverted, much processor power and system resources, such as a memory, are consumed, and cost performance is bad. Although a disk unit can have User Information about the file and file which are stored in the disk unit and can manage, there are problems, such as a synchronization of management information, and realization is easily difficult. The purpose of this invention is to make it possible to restrict a data access effectively by notifying access restriction at the time of the function execution request to a disk unit.

[0004]Other purposes of this invention can respond to various OS's, and making it possible to restrict an easy data access also has mounting of limitation information.

[0005]
[Means for Solving the Problem]A storage for this invention being the disk unit connected to a server and a client apparatus via a network, and memorizing data, in order to attain the above-mentioned purpose, A

means by which it has a control device and this control device controls input and output to said storage, A function which it was transmitted from said server and received, and a means to store function data about this function in a memory. He is trying to have a means to perform this function according to an execution demand of said function from said server, and a limit means which restricts access to data memorized by said storage at the time of this functional execution based on said function data.

[0006]Said function data have a list in which an accessible range is shown, and said limit means is made to restrict access based on this list.

[0007]He is trying for each item of said list to have a lead, a light, and an attribute about access restriction at the time of functional execution of ** which can be performed.

[0008]When access which breaks at restriction of said access occurs, he is trying for said control device to have a means to which abnormal termination of the execution of said function is carried out.

[0009]He is trying for said control device to have a means to return data remembered to be a means to supervise whether execution of said function was performed normally by said storage when execution of said function is not performed normally to a state before performing said function.

[0010]A means by which said control device supervises whether execution of said function was performed normally, A means to set directions of a user of whether to return data memorized by said storage to a state before performing said function as said function data when execution of said function is not performed normally, Only when directions which return data memorized by said storage to a state before performing said function when execution of said function is not performed normally are set as these function data, He is trying to have a means to return data memorized by said storage to a state before performing said function.

[0011]Said control device is kept from overwriting update information based on execution of a function at data before updating until execution of said function is completed.

[0012]He is trying for said control device to store update information based on execution of said function in a memory in this control device.

[0013]It is the server apparatus connected to a client apparatus and a disk unit via a network, A means to generate function data which restrict an access range to data memorized by storage of said disk unit for every function execution request when an execution demand of a function in said disk unit is received from said client, He is trying to have a means to transmit these function data to said disk unit.

[0014]This server apparatus receives user ID information at least from said client apparatus, and he is trying for a means to generate said function data to generate this received function data which restrict said access range based on user ID information at least.

[0015]Are the disk unit connected with a client apparatus via a network, have a storage for memorizing data, and a control device, and this control device, A means to receive a function execution request and user ID information from a client apparatus via a network. He generates function data which restrict an access range to data memorized by said storage for every function execution request based on this user ID information, and is trying to have a means to restrict an access range based on these function data.

[0016]
[Embodiment of the Invention]<<Example 1>> This example is described using a drawing below. One example of the computer system concerning this invention is shown in drawing 1. Although this example shows the disk drive 118 as the disk unit 103, it can constitute also from a subsystem which comprises two or more disk drives 118 as the disk unit 103. As shown in drawing 1, in this example, interconnection of the client 101, the server 102, and the disk unit 103 is carried out in the network 104. Although the server 102 is separately formed in this example, the client 101 and the disk unit 103 with the equivalent function of the server 102 described by this example are also realizable. Namely, what is necessary is just to mount the function server mentioned later in the client 101 or the disk unit 103. The server 102 has CPU105 which executes a program and CPU105 is connected to the memory 111 via the internal bus 106 and the memory control part 109. The program executed by CPU105 and data required at the time of execution are stored in the memory 111. In this example, the function data 127 and the function server 113 are stored on the memory 111, and the function 112 is stored on the disk unit 107. Except the execution time, the function 112, the function server 113, and the function data 127 are stored in the disk unit 107, and if needed, CPU105 takes out directions to the disk control section 108 connected to the internal bus 106, takes them out from the disk unit 107, and uses them. The function data 127 hold the control information about each function 112. In this example, the data written in the disk unit 107 and the data read from the disk unit 107 are stored on a memory. The server 102 is connected to the network 104 via the network control section 110, and CPU105 controls the network control section 110 via the internal bus 106.

[0017]The disk unit 103 has CPU115 which processes, and the drive I/F control section 117 reads the data used on the program executed by CPU115, and a program from the drive 118 with directions of CPU, and it stores it on the memory 120. The memory control part 119 connects with CPU115 via the internal bus 116, and performs access control to access to the memory 120 from CPU115. The functional manager 123, the functional scheduler 125, the function 122, the access range list 124, and the function data 126 are stored in the memory 120. The function data 126 hold the information about the function 122, and the information on the access range list 124 that the access range of the function 122 is restricted. The data of the disk unit 103 is stored in the drive 118. The drive 118 is controlled by the drive I/F control section 117 connected with CPU115 via the internal bus 116. The data exchanged with the drive 118 is stored in memory top 120 by the drive I/F control section 117 by directions of CPU115. The disk unit 103 is connected in the network 104 in the client 101 and the server 102, and the network 104 is controlled by the network control section 121 with directions of CPU115. The data exchanged by network 104 course is stored on the memory 120.

[0018]One example of the program structure of the server 102 concerning this example is shown in drawing 2. The server 102 has the file system 202 in the operating system 201, and mounts the function server 113 on the operating system 201. The function server 113 manages the function 112 performed by the disk unit 103 using the function data 127, and when the function server 113 is required, it reads the function 112 into the memory 111 on the server 102 from the function list 204 of the disk unit 107. The file system 202 manages attributes, such as arrangement of the data stored in the disk unit 107, and User Information.

[0019]Drawing 3 shows one example of the function data 127 concerning this example. The function data 127 are tables which store the management information of the function 112 in the server 102. ID301 which is a management number, the name 302, the owner of a function, The execution authority and the access permission of a function. It comprises the accessing object zero to n-1308 which shows the shown attribute 303, the off-road finishing disk unit 303 which shows whether it is finishing [function / off-road one] already, the bar SHON information 305 on a function, the size 306 which shows the size of a function, the execution level 307 of a function, and the access file at the time of functional execution. Accessing object zero to n-1 308 shows the file which the file system 202 manages, and includes the information on the name of a file, size, and an attribute. The above-mentioned management information is fundamentally set up by the user at the time of registration of a function. The accessing object 308 can be set up by a user also at the time of execution of a function.

[0020]Drawing 4 shows one example of the program structure of the disk unit 103 concerning this example. The drive 118 is controlled by the disk unit 103 using the drive control program 402 of the operating system 401. The function 404 sent from the server 102 is stored in the drive 118, and it reads on the memory 120 if needed. The functional manager 123 is mounted on the operating system 401. The functional manager 123 has the functional scheduler 125 which performs scheduling of the function 122 under execution, and the functional scheduler 125 manages each function 122 at the execution cue 403. The execution cue 403 performs each function 122 according to the priority of the function 122. In this example, the function 122 of a priority linked to the priority 1 is the highest, and the priorities 2, 3, and 4 and a priority become low. The function 122 to perform is connected with the execution cue 403, and the function 122 of the same priority is tied by the list. Function data 00, 01, 02, 03, and 04 126 is the management information of the function 122 and calls it the function data 126. Each function data 126 hold the access range list 124 in which an accessible field is shown.

[0021]Drawing 5 is a flow chart showing operation of the functional manager 123 on the disk unit 103. After starting of the disk unit 103, the functional manager 123 is read from the drive 118 on the memory 120, and is performed. The functional manager 123 prepares the field of the execution cue 403, the function data 126, and the access range list 124 first (501). Next, when waiting and a demand come that a demand comes, demand classification is analyzed (503), functional off-road processing (507) is performed at the time of functional off-road one, and, in the execution demand (504) of a function, functional executive operation (506) is performed. It is reported that he has no applicable demand in other than the above-mentioned 2 demands (505). Repeat execution of the above processing is carried out.

[0022]Drawing 6 shows one example of the management information of the function 122 in the disk unit 103 concerning this example. The function data 126 of drawing 6 are created by the server 102 based on the function data 127, and are sent to the disk unit 103 with the execution demand of a function. The function data 126 the version of ID601 which is a management number of a function, the name 602, the pointer 603 in which the functional storing position on the drive 118 is shown, the size 604 which shows the size of the function 122, the execution level 605 which shows the priority at the time of execution,

and the function 122. The following pointer 607 (by a diagram) for pointing to the following function, when it connects with the version 606 and the execution cue 403 which are shown According to the example of drawing 4, the pointer 608 in which the place of the access range list 124 described to be a start address of the function 00 is shown, the size 609 which shows the size of a list, and 610 elements under list are included.

[0023]Drawing 7 shows one example of the access range list 124 concerning this example. The access range list 124 of drawing 7 is created by the server 102 based on the function data 127, and is sent to the disk unit 103 with the execution demand of a function. Based on the accessing object 308 of the function data 127, an attribute (reading (r), writing in (w), (x) which can be performed) is acquired from the file system 202 with the physical address of this accessing object 308 (a directory or a file), and, specifically, the access range list 124 is created. The access range list 124 has one or more items, and attribute 703** which shows the head 701 which shows the start address of an access range, the size 702 which shows a size, and the attribute of data is contained in each item. An attribute is read and has three sorts of (x) which can be performed [(r), writing (w), and]. When it can perform, it can perform from on a memory as a program. Although this example showed the physical address, it can specify also by management ID of an object as shown in NASD.

[0024]The operation which transmits a function to the disk unit 103 first is explained.

<Functional transmission> Drawing 8 is used first and transmission of a function is explained. Drawing 8 is a figure showing the exchange of the information at the time of normal of functional transmission.

Although this example shows the procedure which transmits a function from the client 101, the transmitting origin of a function is feasible also except client 101. The client 101 requires functional transmission from the function server 113 first (801). At this time, User Information of a transmitting agency is also transmitted to the function server 113 at a demand. The function server 113 receives a demand (802), analyzes a demand, and checks a user's authority etc., and the functional manager 123 on the disk unit 103 of a transmission destination is asked to whether to be functional ability ready for sending (803). If it is ability ready for sending about a function, it will be notified to the client 101 that it is ability ready for receiving (804). If the client 101 receives that it is functional ability ready for sending (805), it will transmit the information about a function to the server 102 first (806). The server 102 receives information (807). This information is used in order to create the function data 127 on the function server 113. The function server 113 creates the function data 126 sent to a client, after creating the function data 127 on a function server from the information received from the client 101 (808).

[0025]The contents of the function data 127 have the transmission destination disk unit 103, a name of a function, size, an execution level, an accessing object, etc. An accessing object specifies a file and a directory. The physical address on the disk unit 103 which corresponds to the information which received the function server 113 which received information from the client 101 at a use file, And an attribute is acquired from a file system and an operating system, the function data 126 and the access range list 124 are created, and it sends to a client. Using the value of the function data 127 of the function server 113, ID601, the name 602, the execution level 605, and the version 606 are the physical addresses on the disk unit 103 in which this function is stored, and acquire the pointer 603 and the size 604 from the file system 202. Access range list information stores the information about the access range list 124. The pointer 608 the position information over the access range list 124. The shown pointer 608, the size 609 which shows the size of the access range list 124, and 610 elements which show the number of the elements of an access range list are created after creation of the access range list 124 shown below based on the information at the time of creation.

[0026]The preparation method of the access range list 124 is explained using drawing 9 and drawing 10. Signs that the data of the result file which the operating system and the file system were asked by a diagram is arranged at the position of the data 0 and the position of the data n-1 are shown. As shown in a figure, when one file is fragmented (arranged apart from each other), It is each consecutive part data 0 like drawing 9, 902 and data n-1 How to specify 903 for the information 906 and the information 907 during a list individually in the access region 904 and the access region 905. They are the initial data 0 of the disk unit 103 like drawing 10, 1001 and end data n-1 There is a method of specifying the information 1004 as a list in the access region 1003 containing that of 1002. In order to set up finely, to instead of [with sufficient reliability], the access range list 124 of drawing 9 is large, in drawing 10, in order to set up roughly, reliability worsens, but the size of the access range list 124 becomes small. It is dependent on a system which method is adopted. When it mounts this method, it is desirable for a file system to be conscious and to store a file in a continuation field.

[0027]After it receives information, to the client 101, the function server 113 notifies receipt (809), and

the client 101 receives function-data receipt (810), and, subsequently to the server 102, it transmits a function (811). If a function is received from the client 101 (812), the function server 113 will store this function in the disk unit 103, and will register information to the function list on the function server 113 (813). Next, the server 102 doubles an access range list and a function with the function data 126, and transmits this doubled information to the disk unit 103 specified from the client 101 (814). If this information is received from the function server 113 (815), the disk unit 103 will be registered into a function list (816), and will transmit a sending end and a storing result to the function server 113 (817). The function server 113 receives a result (818), and transmits a result similarly to the client 101 (819), and the client 101 receives this (820).

[0028] Operation of the client 101 is shown in detail using drawing 11. The user to whom the client 101 operates the client 101 first transmits the issue requesting of a function to the function server 113 (1101). A reception report is received from the function server 113 (1102), next the transmission destination disk unit 103 required at the time of functional transmission, the name of a function, a version, size, an execution level, an accessing object, etc. are transmitted to the function server 113 (1103). Next, a reception report is received, and since it is a functional transmission error if it is an error, error handling is performed and it ends (1110). Processing will be continued if normal. When a reception report (1104, 1105) is analyzed and there is no function on the function server 113, transmit a function (1106-1109). After the report from the server 102 is normally transmitted to a function by waiting and the disk unit 103, without already transmitting on the function server 113 in a certain case, if functional transmission is normal, processing will be ended as it is, and if it is an error, deed processing will be finished for error handling (1110).

[0029] Operation of the function server 113 is shown in detail using drawing 12. The function server 113 receives issue of a function from the client 101 first (1201). It transmits having received, when the function investigated in ability ready for sending (1202) and was ready for sending to the client 101 (1203), and processing is continued, if the error has occurred, an error generation will be notified to the client 101, error generation processing is performed, and processing is finished (1221). Next, the function data 127 are received (1204, 1205). The function data 127 transmitted first are checked at this time. When there is already no function concerned of the same version in the function server 113, a function is received from the client 101 (1207-1210). When there is a function, functional transmitting processing (1211-1220) to a disk unit is performed.

[0030] a function is received from a client — it processing(1207-1210) — attaches and explains. 1207, 1208 which receive a function from the client 101 when this function cannot be found on the function server 113. When it tells having received to the client 101 when it receives normally (1210), and an error occurs, an error is reported and processing is ended (1221). Next, the received function is stored on a memory. Next, a function is stored in the disk unit 107. The function server 113 is stored using the file system 202 on the server 102. The file name at this time decides that it is decided on a system that the function server 113 will be a meaning. ID of a function is also determined that it is decided on a system that it will be a meaning. The user who transmitted this demand from the client 101 is made into an owner, the attribute execution, reference, and deletion of whom are possible in a function is created, the name, the version, the size, execution level, and accessing object which were further transmitted from the client 101 are added, and the function data 127 of this function are created. The created function is stored and saved at the function list 204.

[0031] Next, the functional transmission to the functional manager 123 of the disk unit 103 is explained (1211-1220). First, it investigates [ending with transmitting, or] to the disk unit 103 (1211). Case [which has already been transmitted to the disk unit 103], the notice of a sending end is transmitted to the client 101, and processing is finished (1220). When a function has not transmitted, functional transmitting processing to the disk unit 103 is performed henceforth. The Request to Send of a function is first published and carried out to the disk unit 103 (1212). Next, transmission of the function data 126 and a function is performed from the function server 113 to the disk unit 103. The function data 126 hold the information on the access range list 124. First, the demand of functional transmission is published (1212) and then the function data 126 are transmitted (1214). The function server 113 transmits the function data 126 to the functional manager 123 of the disk unit 103 (1214). When the function data 126 are transmitted normally, transmission (1217) of a function is performed. When transmission of the function data 126 becomes an error, an error is reported to the client 101 and processing is finished (1216, 1221). After transmitting the function data 126 normally, the function server 113 transmits a function to the functional manager 123. When normal transmission is reported to a client when a function is transmitted normally (1220), and transmission of a function is normal, after registering this disk unit 103 into the

transmitted disk unit 103 of the function data 126, transmitting processing normal termination is notified to the client 101, and processing is finished. When an error occurs at the time of transmission of a function, the function and the function data 126 on the function server 113 hold the transmission to the disk unit 103 in the state of un-transmitting at this time.

[0032] The functional manager's 123 operation is shown in detail using drawing 13. The demand of functional transmission to the functional manager 123 is transmitted to the beginning from the function server 113 (1301). When a function cannot be received, the functional manager 123 notifies a purport [being unreceivable] to the function server 113, and finishes processing (1302, 1311). When a function is receivable, it reports that functional reception is possible to the function server 113, and processing is continued henceforth (1302, 1303). Next, the functional manager 123 waits to transmit the function data 126 from the function server 113 (1304). The functional manager 123 will change into the function data 126 of the functional manager 123 on the disk unit 103 based on the function data 126 transmitted first, if the function data 126 are transmitted (1305). Next, the functional manager 123 waits to transmit a function from the function server 113 (1306). If a function is sent, the field of the size of the function described at the function data 126 will be secured, and it will write in the drive 118 (1307). The start address at this time is added to the function data 126. If a function is stored normally, the function data 126 will be stored in the arbitrary fields of the drive 118, the purport of functional normal reception will be notified to the function server 113, finally (1308, 1309) a function will be registered into the function list 405, and processing (1310) will be finished.

[0033] <Functional execution> The rough flow of processing is first explained using drawing 14. The disk unit 103 which performs the function which the user of the client 101 wants to perform from now on, and a function is notified to the function server 113 (1401). It is reported that a request content is analyzed and the function server 113 investigates the existence etc. of the demanded user authority (1403), and it can be performed when it can perform if the demand from the client 101 is received (1402) (1404). If it thinks that the client 101 can be performed (1405), a parameter required for functional execution will be transmitted to the function server 113 (1406). Here, parameters are a file name to be used and data required at the time of functional execution. The function server 113 receives a parameter (1407), takes out the function data 126 associated to this function from the function list 204, and creates the function data 126 transmitted to the functional manager 123 of the disk unit 103, and the access range list 124 (1408). At this time, the field is already specified as the access range list 124, and when a file and a directory are further specified from the client 101, a field is added to the access range list 124. The function server 113 transmits the information created with the execution demand of the function to the functional manager 123 (1409). The functional manager 123 will notify the report of functional reception to the function server 113, if the execution demand of a function is received (1410) (1411). Next, when the functional manager 123 performs the received function (1413) and a function is completed normally, The executed result of an end and a function is notified to the function server 113 (1414), and the function server 113 notifies the executed result of an end and a function to the client 101, after receiving and (1415) analyzing this notice (1416). The client 101 receives the executed result of an end and a function (1417), reports an end to a user, and finishes processing.

[0034] Next, the flow of an execution demand of a function is explained using drawing 15. When performing a function with the user of the client 101, the demand of functional execution is transmitted to the function server 113 at the beginning (1501). The disk unit 103 and user who perform a function with the demand of functional execution are notified. Although it was considered as the disk unit 103 which performs a function here, since it can be decided from the file of the file system 202 on the server 102 that the disk unit 103 will be a meaning, it is feasible even if it specifies the file of the file system on the server 102. The client 101 receives the report of reception of a function from the function server 113, after publishing the execution demand of a function (1502). When a function is registered into a server and there is, transmitting processing of a function is performed (1504) and transmission is performed normally, processing is continued, an error is notified at the time of an error generation, and processing is finished (1512). [no] Next, since a function cannot be performed when the contents of the report cannot be performed, deed processing is ended for error handling (1506, 1512). When a function can be performed, a parameter required [the next] at the time of functional execution is transmitted (1507). Here, parameters are a file to be used, a directory, and data required at the time of functional execution. After transmitting a parameter, the received result of a parameter is received (1508) and, in an error, deed processing is ended for error handling. It continues processing, in being normal. Waiting (1510) and the client 101 receive the executed result of the end of functional, and a function for the end of execution of a function (1511), in the end of an error, it reports that performed error handling and the error occurred

to the user, and it ends processing (1512). When processing is performed normally, a user is notified of the purport and functional executed result of normal processing, and processing is finished (1513).

[0035]Next, the processing at the time of functional execution of the function server 113 is explained using drawing 16. The function server 113 receives the execution demand of a function from the client 101 first (1601). At this time, the client 101 specifies the disk unit 103 which performs a function with a demand, and the user of a requiring agency. The function server 113 searches a request function from the function list 204. By there being no request function in the function list 204, or investigating the function data 126, when this function is not ending with transmitting at this disk unit 103, (1602), a function is transmitted for the function which the client 101 has to the function server 113 to the functional manager 123 by function server 113 course (1603). Processing is continued when functional transmission is completed normally. When a function is not able to transmit normally, it notifies the client 101 that functional transmission went wrong, and processing is finished (1604, 1605). When the function is transmitted to the disk unit 103, the function data 126 of this function are taken out from the function list 204, when you investigate whether there is any authority for a request source user to perform a function to this function (1606) and there is no execution authority of a function, be transmitted to there being no execution authority at the client 101 — processing is finished (1607, 1608). When there is execution authority of a function, it tells that it can perform to the client 101 (1607, 1609).

[0036]Next, a parameter required at the time of functional execution is received from the client 101 (1610), and the parameter about an access range is added to the accessing object of the function data 126. Next, an accessing object is changed. An accessing object acquires a file or the physical address of a directory from the file system 202 and the operating system 201 on the server 102, asks for the physical position of a file, and creates the access range list 124. It is investigated whether there is any access permission to the file and directory to which the request source user was specified at this time (1612). When there is no access permission, a transaction error is notified to the client 101, and processing is finished (1608). Processing is continued when it can access to all the accessing objects.

[0037]Next, the function server 113 transmits parameters other than an access range, and the access range list 124 to the functional manager 123 with the execution demand of a function (1613). When a reception report (1614) of a function execution request is an error, an error is notified to the client 101 and processing is finished (1618). When normal, it waits for a report of the end of execution of a function (1615). After the end of execution of a function, after receiving a report of the end of functional from the functional manager 123, the function server 113 investigates the contents of a report, if it is an error, will notify an error to the client 101 and will finish processing (1616, 1618). If the report is normal, the purport of normal termination and the executed result of a function will be notified to the client 101, and processing will be finished (1616, 1617). As shown above, when the function is not transmitted to the functional manager 123, according to the procedure of transmitting processing of a function, transmitting processing of a function is performed before the execution demand of a function to the functional manager 123.

[0038]A function explains below the procedure which performs a function at the time already transmitted to the functional manager. Operation of the functional manager 123 at the time of functional execution is explained using drawing 17. The functional manager 123 receives a parameter required at the time of execution, and the access range list 124 with the execution demand which receives the execution demand of a function from the function server 113 (1701). The size of an accessible field and a field and an attribute are stored in the access range list 124. This function is inserted in the execution cue 403 after registering into the function data 126 and the access range list 124 the information transmitted from the function server 113. A priority is decided from the execution level of the function data 126, and a function is inserted. The functional manager's 123 functional scheduler 125 is performed in an order from the high function of a priority. The function under execution carries out the end of execution, or it performs by fixed time CPU, or when [such as IO,] waiting occurs without carrying out CPU use, the function under execution is taken out from the execution cue 403, and then the high function of a priority is performed. The function performed by fixed time CPU is again inserted in the last of the priority of the execution cue 403 concerned. The function which became waiting, such as IO, is again inserted in the execution cue 403, after a waiting state is canceled. The function is performed by the scheduling shown above.

[0039]If access to the drive 118 occurs at the time of functional execution (1702), the address of an access point will question the functional manager 123 in within the limits of the access range list 124 (1703). At the time of access to the outside of the range of an access list, it is an error, and the functional manager 123 is notified of the access error outside a permission range, and processing is finished (1709). Access classification is investigated at the time in an access range (1704). When only a

lead is good and the attribute of the data of the light to an access range, etc. to access differs from the contents of access, the functional manager 123 is notified of the access error outside a permission range as similarly as the point, and processing is finished (1709). Processing will be performed if access classification is in a designated range (1705). When a function is normally completed continuously (1707) until execution of a function ends the above-mentioned processing, the functional manager 123 notifies the executed result of normal termination and a function to the function server 113, and finishes processing (1708).

[0040]<<Example 2>> This example 2 is hereafter described using a drawing. The feature of this example is that it does not leave the state in the middle of renewal of a function, but can return before execution of a function (rollback), when an error occurs at the time of execution of a function. One example of the computer system concerning this invention is shown in drawing 18. Although this example shows the disk drive 118 as the disk unit 103, it can constitute also from a subsystem which comprises two or more disk drives 118 as the disk unit 103. As shown in drawing 18, in this example, interconnection of the client 101, the server 102, and the disk unit 103 is carried out in the network 104. Although the server 102 is separately formed in this example, the client 101 and the disk unit 103 with the equivalent function of the server 102 described by this example are also feasible. The server 102 has CPU105 which executes a program and CPU105 is connected to the memory 111 via the internal bus 106 and the memory control part 109. The program executed by CPU105 and data required at the time of execution are stored in the memory 111. In this example, the function data 127 and the function server 113 are stored on the memory 111, and the function 112 is stored on the disk unit 107. Except the execution time, the function 112, the function server 113, and the function data 127 are stored in the disk unit 107, and if needed, CPU105 takes out directions to the disk control section 108 connected to the internal bus 106, takes them out from the disk unit 107, and uses them. The function data 127 hold the control information about each function 112. In this example, all of the data written in the disk unit 107 and the data read from the disk unit 107 are stored on the memory 111. The server 102 is connected to the network 104 via the network control section 110, and CPU controls the network control section 110 via an internal bus.

[0041]The disk unit 103 has CPU115 which processes, and the drive 118 to the drive I/F control section 117 reads the data used on the program executed by CPU115, and a program from the drive 118 with directions of CPU, and it stores it on the memory 120. The functional manager 123, the functional scheduler 125, a function, the access range list 124, the cache memory 1801, and the cache management table 1802 are stored in the memory 120. The memory control part 119 connects with CPU115 via the internal bus 116, and performs access control to access to the memory 120 from CPU115. The functional manager 123, the functional scheduler 125, the function 122, the access range list 124, and the function data 126 are stored in the memory 120. The function data 126 hold the information about the function 122, and the information on the access range list 124 that the access range of the function 122 is restricted. The data of the disk unit 103 is stored in the drive 118. The drive 118 is controlled by the drive I/F control section 117 connected with CPU via an internal bus. The data exchanged with the drive 118 is stored on the memory 120 by the drive I/F control section 117 by directions of CPU115. The disk unit 103 is connected in the network 104 in the client 101 and the server 102, and the network 104 is controlled by the network control section 121 with directions of CPU115. The data exchanged by network 104 course is stored on the memory 120.

[0042]Drawing 19 shows one example of the program structure of the disk unit 103 concerning this example. The drive 118 is controlled by the disk unit 103 with the drive control program 402 in the operating system 401. The function sent from the server 102 is stored in the drive 118, and it reads on the memory 120 if needed. The functional manager 123 is mounted on the operating system 401. The functional manager 123 has the functional scheduler 125 which performs scheduling of the function under execution, and the functional scheduler 125 manages each function at the execution cue 403. The execution cue 403 performs each function according to the priority of a function. In this example, the function of a priority linked to the priority 1 is the highest, and the priorities 2, 3, and 4 and a priority become low. The function to perform is connected with the execution cue 403, and the cue of the same priority is tied by the list. Function data 00, 01, 02, 03, and 04 126 is the management information of the function 122 and calls it the function data 126. Each function data 126 hold the access range list 124 in which an accessible field is shown. The cache memory 1801 and the cache management table 1802 for managing the access information to the drive 118 are mounted in the functional manager 123, and the rollback of a function to the drive 118 is made possible. Although cache memory 1801 was mounted in the memory 120 in this example, since there is a possibility that it may become impossible to be unable to store in the memory 120 when update information becomes extensive, it is also possible to prepare the

cache memory for update information for the drive 118. The function data 126 of the function server 113 and the function server 113 are the same as that of Example 1.

[0043]Drawing 20 shows one example of the function data 126 concerning this example. The owner of ID301, the name 302, and the function in which the function data 126 are the management numbers of a function, The execution authority and the access permission of a function. the shown attribute 303 — a function already. Whether it is finishing [off-road one]. The shown off-road finishing disk unit 303, the file name 304 which is the functional names on a file system, the bar SHON information 305 on a function, the rollback flag 2001 which shows directions of the rollback of a function, the size 306 which shows the size of a function, the execution level 307 of a function, Accessing object zero to n-1 which shows the access file at the time of functional execution It comprises 308. Accessing object zero to n-1 308 shows the file which the file system 202 manages, and includes the information on the name of a file, size, and an attribute. The access range list 124 is the same as that of Example 1. Transmitting processing of a function is performed like Example 1.

[0044]Next, the flow of execution of a function is explained.

<Functional execution> Next drawing 21 is used and the flow of an execution demand of a function is explained. When performing a function with the user of the client 101, the demand of functional execution is communicated to the function server 113 at the beginning (2101). The disk unit 103 and user who perform a function with the demand of functional execution are notified. Although it was considered as the disk unit 103 which performs a function here, since it can be decided from the file of the file system 202 on the server 102 that the disk unit 103 will be a meaning, it is feasible even if it specifies the file of the file system 202 on the server 102. The client 101 receives the report of reception of a function from the function server 113, after publishing the execution demand of a function (2105). When the contents of the report cannot perform a function, since a function cannot be performed, it ends deed processing for error handling. When a function can be performed, a parameter required [the next] at the time of functional execution is transmitted (2106).

[0045]Here, parameters are directions of the file to be used, a directory, data required at the time of functional execution, and a rollback. Differing from Example 1 is in a parameter at the point which carries out rollback directions. In not carrying out rollback directions, even if an error occurs in the middle of a function, the data in the middle of execution remains still in the state in the middle of execution of a function. When there are directions of a rollback and an error occurs in the middle of execution of a function, data returns to the state before execution of a function.

[0046]After transmitting a parameter, the received result of a parameter is received and, in an error, deed processing is ended for error handling. It continues processing, in being normal. Waiting and the client 101 receive the executed result of the end of functional, and a function for the end of execution of a function (2117), in the end of an error, it reports that performed error handling and the error occurred to the user, and it ends processing. When processing is performed normally, a user is notified of the purport and functional executed result of normal processing, and processing is finished. Since it is the same as that of the explanation about drawing 14, the explanation about functional execution of a function server and a functional manager is omitted.

[0047]Next, the flow of an execution demand of a function is explained using drawing 22. When performing a function with the user of the client 101, the demand of functional execution is transmitted to the function server 113 at the beginning (2201). The disk unit 103 and user who perform a function with the demand of functional execution are notified. Although it was considered as the disk unit 103 which performs a function here, since it can be decided from the file of the file system 202 on the server 102 that the disk unit 103 will be a meaning, it is feasible even if it specifies the file of the file system on the server 102. The client 101 receives the report of reception of a function from the function server 113, after publishing the execution demand of a function (2202). When a function is registered into a server and there is, transmitting processing of a function is performed (2204) and transmission is performed normally, processing is continued, an error is notified at the time of an error generation, and processing is finished (2212). [no]

[0048]Next, since a function cannot be performed when the contents of the report cannot be performed, deed processing is ended for error handling (2206, 2212). When a function can be performed, a parameter required [the next] at the time of functional execution is transmitted (2207). Here, parameters are directions of the file to be used, a directory and data required at the time of functional execution, and a rollback. After transmitting a parameter, the received result of a parameter is received (2208) and, in an error, deed processing is ended for error handling. It continues processing, in being normal. Waiting (2210) and the client 101 receive the executed result of the end of functional, and a function for the end of

execution of a function (2209), in the end of an error, it reports that performed error handling and the error occurred to the user, and it ends processing (2212). When processing is performed normally, a user is notified of the purport and functional executed result of normal processing, and processing is finished (2213).

[0049]Next, the processing at the time of functional execution of the function server 113 is explained using drawing 23. The function server 113 receives the execution demand of a function from the client 101 first (2301). At this time, the client 101 specifies the disk unit 103 which performs a function with a demand, and the user of a requiring agency. The function server 113 searches a request function from the function list 204. By there being no request function in the function list 204, or investigating the function data 127, when this function is not ending with transmitting at this disk unit 103, (2302), A function is transmitted for the function which the client 101 has to the function server 113 to the functional manager 123 by function server 113 course (2303). Processing is continued when functional transmission is completed normally. When a function is not able to transmit normally, it notifies the client 101 that functional transmission went wrong, and processing is finished (2304, 2305). When the function is transmitted to the disk unit 103, the function data 126 of this function are taken out from the function list 204, It investigates whether there is any authority for a request source user to perform a function to this function (2306), and when there is no execution authority of a function, tradition processing is finished for the purport of execution authority to the client 101 (2307, 2308). When there is execution authority of a function, it tells that it can perform to the client 101 (2307, 2309).

[0050]Next, a parameter required at the time of the functional execution including directions of a rollback is received from the client 101 (2310), and the parameter about an access range is added to the accessing object of the function data 127. Next, an accessing object is changed. An accessing object acquires a file or the physical address of a directory from the file system 202 and the operating system 201 on the server 102, asks for the physical position of a file, and creates the access range list 124. It is investigated whether there is any access permission to the file and directory to which the request source user was specified at this time (2312). When there is no access permission, a transaction error is notified to the client 101, and processing is finished (2308). Processing is continued when it can access to all the accessing objects.

[0051]Next, the function server 113 transmits the parameter and the access range list 124 which include the rollback directions of those other than an access range to the functional manager 123 with the execution demand of a function (2313). When a reception report (2314) of a function execution request is an error, an error is notified to the client 101 and processing is finished (2318). When normal, it waits for a report of the end of execution of a function (2315). After the end of execution of a function, after receiving a report of the end of functional from the functional manager 123, the function server 113 investigates the contents of a report, if it is an error, will notify an error to the client 101 and will finish processing (2316, 2318). If the report is normal, the purport of normal termination and the executed result of a function will be notified to the client 101, and processing will be finished (2316, 2317). As shown above, when the function is not transmitted to the functional manager 123, according to the procedure of transmitting processing of a function, transmitting processing of a function is performed before the execution demand of a function to the functional manager 123.

[0052]A function explains below the procedure which performs a function at the time already transmitted to the functional manager. Operation of the functional manager 123 at the time of functional execution is explained using drawing 24. The functional manager 123 receives a parameter required at the time of execution, rollback directions, and the access range list 124 with the execution demand which receives the execution demand of a function from the function server 113 next (2401). The size of an accessible field and a field and an attribute are stored in the access range list 124. This function is inserted in the execution cue 403 after registering into the function data 126 and the access range list 124 the information transmitted from the function server 113. Scheduling of the execution cue 403 is performed like Example 1. The processing after being inserted in the execution cue 403 is explained using drawing 24. The existence of rollback directions is investigated first (2402), if there is rollback specification, a rollback flag will be set to ON (2404), and a rollback flag will be set to OFF if there is nothing (2403). Next, when there is no access to the drive 118 during execution of processing, it performs as it is, and when there is access to the drive 118, the check of an access range and the check of access classification are performed like Example 1 (2406, 2407). When it becomes an error at the time of this check, roll back processing is performed. Roll back processing is performed as follows. When a rollback flag is ON, after canceling the non-update information on (2412) and the cache memory 1801 (2413), an error is notified to the functional manager 123 (2414). When a rollback flag is OFF, after writing the update information on the

cache memory 1801 in the drive 118 (2415), an error is notified to the functional manager 123 (2414). Processing is performed when an access check is normal (2408). When an error occurs as a result of processing execution, the above-mentioned roll back processing is performed.

[0053]When execution of a function is completed normally, the non-update information on the cache memory 1801 is written in the drive 118, and the functional manager 123 is notified of the normal termination of a function.

[0054]Next, processing of IO to the drive 118 at the time of functional execution is explained using drawing 25. In access to the drive 118, this processing is performed at the time of functional execution. The functional manager 123 investigates the processing classification to the drive 118 first (2501). In the case of a light, when data is on the cache memory 1801, data overwrite of the new updating is carried out to old data. When data does not exist on the cache memory 1801, on the cache memory 1801, a field is newly secured and data is stored to the secured field (2502, 2503, 2504). When the contents of the cache memory 1801 are updated at the time of a light, the cache management table 1802 is updated (2505). When the access classification to the drive 118 is a lead and data exists on the cache memory 1801, the data on the cache memory 1801 is used as lead data (2507). In not existing on the cache memory 1801, the data on the drive 118 is used as lead data (2508), and it continues processing. Drawing 26 illustrates the case where it leads from a drive, drawing 27 illustrates the case where a light is carried out to cash, drawing 28 illustrates the case where it leads from cash, and drawing 29 illustrates the case where a light is carried out to a drive.

[0055]

[Effect of the Invention]According to this invention, a data access can be effectively restricted to the data access at the time of the functional execution in a disk unit.

[0056]In various OS's, restriction of a data access can be performed similarly, and mounting of limitation information and mounting can be performed easily.

[Translation done.]

* NOTICES *

JP0 and INPIT are not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.*** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]

- [Drawing 1]It is a figure showing the composition of the whole in Example 1 of this invention.
- [Drawing 2]It is a figure showing the composition of a server.
- [Drawing 3]It is a figure showing the function data of the function of a server.
- [Drawing 4]It is a figure showing the composition of a disk unit.
- [Drawing 5]It is a figure showing the flow chart of a functional manager's processing.
- [Drawing 6]It is a figure showing the function data of the function of a disk unit.
- [Drawing 7]It is a figure showing the access range list of functions of a disk unit.
- [Drawing 8]It is a figure showing the flow of the processing at the time of normal of the whole functional transmitting processing.
- [Drawing 9]It is a figure showing an example of the specification method of an access range.
- [Drawing 10]It is a figure showing an example of the specification method of an access range.
- [Drawing 11]It is a figure showing the flow chart of the functional transmitting processing in a client.
- [Drawing 12]It is a figure showing the flow chart of the functional transmitting processing in a function server.
- [Drawing 13]It is a figure showing the flow chart of the functional transmitting processing in a functional manager.
- [Drawing 14]It is a figure showing the flow of the processing at the time of normal of the whole functional executive operation.
- [Drawing 15]It is a figure showing the flow chart of the functional executive operation in a client.
- [Drawing 16]It is a figure showing the flow chart of the functional executive operation in a function server.
- [Drawing 17]It is a figure showing the flow chart of the functional executive operation in a functional manager.
- [Drawing 18]It is a figure showing the composition of the whole in Example 2 of this invention.
- [Drawing 19]It is a figure showing the composition of a disk unit.
- [Drawing 20]It is a figure showing the function data of the function on a disk unit.
- [Drawing 21]It is a figure showing the flow of the processing at the time of normal of the whole functional executive operation.
- [Drawing 22]It is a figure showing the flow chart of the functional executive operation in a client.
- [Drawing 23]It is a figure showing the flow chart of the functional executive operation in a function server.
- [Drawing 24]It is a figure showing the flow chart of the functional executive operation in a functional manager.
- [Drawing 25]It is a figure showing the flow chart of IO processing to the drive in a functional manager.
- [Drawing 26]It is a figure showing operation of the lead from a drive.
- [Drawing 27]It is a figure showing operation of the light to cash.
- [Drawing 28]It is a figure showing operation of the lead from cash.
- [Drawing 29]It is a figure showing operation of the light to a drive.
- [Description of Notations]
- 101 Client
- 102 Server
- 103 Disk unit
- 104 Network

- 105, 115 CPU
- 106, 116 internal buses
- 107 Disk unit
- 108 Disk control section
- 109 Memory control part
- 110, 121 network control sections
- 111 and 120 Memory
- 112, 122, and 404 Function
- 113 Function server
- 114 Disk controller
- 117 Drive I/F control section
- 118 Drive
- 123 Functional manager
- 124 Access range list
- 125 Functional scheduler
- 126 and 127 Function data
- 1801 Cache memory
- 1802 Cache management table
- 201 and 401 Operating system
- 202 File system
- 204, 405 function lists
- 402 Drive control program
- 403 Execution cue

[Translation done.]

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開2001-5614

(P2001-5614A)

(43)公開日 平成13年1月12日(2001.1.12)

(51)Int.Cl. ⁷	識別記号	F I	テーマコード*(参考)
G 0 6 F 3/06	3 0 1	G 0 6 F 3/06	3 0 1 A 5 B 0 1 7
	3 0 4		3 0 4 H 5 B 0 6 5
12/00	5 3 7	12/00	5 3 7 M 5 B 0 8 2
	5 4 5		5 4 5 F
12/14	3 1 0	12/14	3 1 0 K
審査請求 未請求 請求項の数11 O L (全 33 頁) 最終頁に続く			

(21)出願番号 特願平11-179620

(22)出願日 平成11年6月25日(1999.6.25)

(71)出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72)発明者 渡邊 直企

東京都国分寺市東恋ヶ窪一丁目280番地

株式会社日立製作所中央研究所内

(72)発明者 高本 良史

東京都国分寺市東恋ヶ窪一丁目280番地

株式会社日立製作所中央研究所内

(74)代理人 100099298

弁理士 伊藤 修 (外1名)

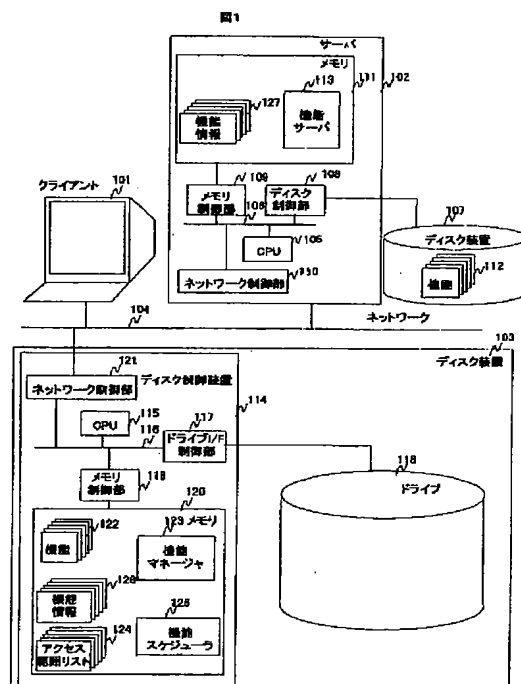
最終頁に続く

(54)【発明の名称】 ディスク装置およびサーバ装置

(57)【要約】

【課題】 ディスク装置に対する機能実行要求時にアクセス制限を通知することで効果的にデータアクセスの制限を行うことを可能とすることにある。

【解決手段】 クライアント101はサーバ102に実行する機能を送信し、同時に送信先ディスク装置、機能名称、バージョン等の情報を送信する。サーバは受信した機能を格納し、受信した情報を用いて機能情報127を作成し、さらにディスク装置103に送るアクセス範囲リスト124を有する機能情報126を作成し、ディスク装置に機能と機能情報126を送信する。ディスク装置はこれらを格納する。クライアントから機能実行要求がサーバに送られると、サーバはパラメータも送られてくれば機能情報126をさらに作成し、ディスク装置に機能実行要求を出すと共に該機能情報126を送る。ディスク装置は機能の実行を行い、実行時のアクセス制限を機能情報126に従って行う。



1

【特許請求の範囲】

【請求項 1】 ネットワークを介してサーバとクライアント装置に接続されたディスク装置であって、データを記憶するための記憶媒体と、制御装置を備え、該制御装置は、前記記憶媒体への入出力を制御する手段と、

前記サーバから送信され、受信した機能と該機能に関する機能情報をメモリに格納する手段と、

前記サーバからの前記機能の実行要求に応じて該機能を実行する手段と、

該機能実行時の前記記憶媒体に記憶されているデータに対するアクセスを前記機能情報に基づき制限する制限手段を有することを特徴とするディスク装置。

【請求項 2】 請求項 1 記載のディスク装置において、前記機能情報はアクセス可能な範囲を示すリストを有し、前記制限手段は該リストに基づきアクセスの制限をすることを特徴とするディスク装置。

【請求項 3】 請求項 2 記載のディスク装置において、前記リストの各項目は、リード、ライト、実行可能等の機能実行時のアクセス制限に関する属性を有することを特徴とするディスク装置。

【請求項 4】 請求項 2 記載のディスク装置において、前記制御装置は、前記アクセスの制限に違反するアクセスが発生した場合に前記機能の実行を異常終了させる手段を有することを特徴とするディスク装置。

【請求項 5】 請求項 1 乃至請求項 4 のいずれかの請求項記載のディスク装置において、前記制御装置は、前記機能の実行が正常に行われたか否かを監視する手段と、

前記機能の実行が正常に行われなかった場合に前記記憶媒体に記憶されているデータを前記機能を実行する前の状態に戻す手段を有することを特徴とするディスク装置。

【請求項 6】 請求項 1 乃至請求項 4 のいずれかの請求項記載のディスク装置において、前記制御装置は、前記機能の実行が正常に行われたか否かを監視する手段と、

前記機能の実行が正常に行われなかった場合に前記記憶媒体に記憶されているデータを前記機能を実行する前の状態に戻すか否かのユーザの指示を前記機能情報に設定する手段と、

前記機能の実行が正常に行われなかった場合に、前記記憶媒体に記憶されているデータを前記機能を実行する前の状態に戻す指示が該機能情報に設定されている場合のみ、前記記憶媒体に記憶されているデータを前記機能を実行する前の状態に戻す手段を有することを特徴とするディスク装置。

【請求項 7】 請求項 5 または請求項 6 記載のディスク装置において、前記制御装置は、前記機能の実行が終了するまで、機能

2

の実行による更新データを更新前データに上書きしないことを特徴とするディスク装置。

【請求項 8】 請求項 7 記載のディスク装置において、前記制御装置は、前記機能の実行による更新データを該制御装置内のメモリに格納することを特徴とするディスク装置。

【請求項 9】 ネットワークを介してクライアント装置とディスク装置に接続されたサーバ装置であって、前記クライアントから前記ディスク装置における機能の実行要求を受け付けた場合、機能実行要求毎に前記ディスク装置の記憶媒体に記憶されているデータに対するアクセス範囲を制限する機能情報を生成する手段と、該機能情報を前記ディスク装置に送信する手段を有することを特徴とするサーバ装置。

【請求項 10】 請求項 9 記載のサーバ装置において、該サーバ装置は前記クライアント装置から少なくともユーザ ID 情報を受信し、

前記機能情報を生成する手段は、該受信した少なくともユーザ ID 情報を基にして前記アクセス範囲を制限する機能情報を生成することを特徴とするサーバ装置。

【請求項 11】 ネットワークを介してクライアント装置と接続されたディスク装置であって、

データを記憶するための記憶媒体と、制御装置を備え、該制御装置は、ネットワークを介してクライアント装置から機能実行要求とユーザ ID 情報とを受信する手段と、

該ユーザ ID 情報に基づき機能実行要求毎に前記記憶媒体に記憶されているデータに対するアクセス範囲を制限する機能情報を生成し、該機能情報に基づきアクセス範囲を制限する手段を有することを特徴とするディスク装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、ネットワークに直接接続したディスク装置に係り、特に該ディスク装置に対し機能をオフロードし該機能を実行するときのデータ保護を可能にするディスク装置および関連するサーバ装置に関する。

【0002】

【従来の技術】本発明の実施例の図 1 を流用して以下説明する。LSI の高集積化技術の向上により、ディスク装置 103 にも従来よりも高性能なプロセッサや高機能な制御 LSI の搭載が可能となってきた。そこで、ディスク装置 103 にネットワーク 104 I/F を設けサーバ 102 だけでなくクライアント 101 からも直接アクセス可能とし、システム全体のスループットを向上させるネットワーク 104 接続ディスク装置 103 が提案されている。CMU (Carnegie Mellon University) の Garth A. Gibson らが提案している NASD (Network-Attached Secure Disks) がある。ACM International Conference on Measurement and Modeling of

Computer Systems(Sigmetrics '97)、Seattle、Washington、June 15-18、1997.における「File Server Scaling with Network-Attached Secure Disks」に詳しい。またディスク装置103へサーバ102の行っている処理をオフロードしサーバ102の負荷を削減しシステムスループットを向上しようというインテリジェントなディスク装置103が提案されており、CMUのErik Riedel、Garth Gibsonらが提案しているActive Disksがある。Active DisksはThe conference paper "Active Storage For Large-Scale Data Mining and Multimedia," Proc. of the 24th International Conference on Very large Database s (VLDB '98)、New York、New York、August 24-27、1998.に詳しい。

【0003】

【発明が解決しようとする課題】ディスク装置103に機能をオフロードし実行する際には、実行時の実行権限、データのアクセス権限等制限は難しい。ここで機能とは、データベースにおけるクライアントからの問い合わせに対する選択／抽出処理、サーバを経由しないクライアントーディスク装置間の直接データの転送、ディスク装置間のデータの2重化、ECU⇔JIS等のデータ変換等があり、サーバにおいて行われている処理を切り出してディスク装置上で実行可能とした処理である。Unixで用いられているようなRPC(Remote Procedure Call)で使用しているような既存の遠隔手続き処理に従った管理を採用可能であるが、OS、ベンダ等が限られてしまうため異種のOSへの対応が難しい。現状ではUnixのRPC(Remote Procedure Call)がもっとも近い形態では在るが、RPCを流用するとファイルを開くたびにファイルのアクセス権限をサーバに取得する必要が在り、多くのプロセスサーバパワーと、メモリ等のシステム資源を消費しコストパフォーマンスが悪い。また、ディスク装置がディスク装置に格納されているファイルとファイルに関するユーザ情報を持ち、管理可能であるが、管理情報の同期等の問題があり容易に実現は困難である。本発明の目的は、ディスク装置に対する機能実行要求時にアクセス制限を通知することで効果的にデータアクセスの制限を行うことを可能とすることにある。

【0004】本発明の他の目的は、各種OSに対応可能であり、制限情報の実装も容易なデータアクセスの制限を行うことを可能とすることにある。

【0005】

【課題を解決するための手段】上記目的を達成するため、本発明は、ネットワークを介してサーバとクライアント装置に接続されたディスク装置であり、データを記憶するための記憶媒体と、制御装置を備え、該制御装置は、前記記憶媒体への入出力を制御する手段と、前記サーバから送信され、受信した機能と該機能に関する機能情報をメモリに格納する手段と、前記サーバからの前記機能の実行要求に応じて該機能を実行する手段と、該機

能実行時の前記記憶媒体に記憶されているデータに対するアクセスを前記機能情報に基づき制限する制限手段を有するようにしている。

【0006】また、前記機能情報はアクセス可能な範囲を示すリストを有し、前記制限手段は該リストに基づきアクセスの制限をするようにしている。

【0007】また、前記リストの各項目は、リード、ライト、実行可能等の機能実行時のアクセス制限に関する属性を有するようにしている。

10 【0008】また、前記制御装置は、前記アクセスの制限に違反するアクセスが発生した場合に前記機能の実行を異常終了させる手段を有するようにしている。

【0009】また、前記制御装置は、前記機能の実行が正常に行われたか否かを監視する手段と、前記機能の実行が正常に行われなかった場合に前記記憶媒体に記憶されているデータを前記機能を実行する前の状態に戻す手段を有するようにしている。

20 【0010】また、前記制御装置は、前記機能の実行が正常に行われたか否かを監視する手段と、前記機能の実行が正常に行われなかった場合に前記記憶媒体に記憶されているデータを前記機能を実行する前の状態に戻すか否かのユーザの指示を前記機能情報に設定する手段と、前記機能の実行が正常に行われなかった場合に、前記記憶媒体に記憶されているデータを前記機能を実行する前の状態に戻す指示が該機能情報に設定されている場合のみ、前記記憶媒体に記憶されているデータを前記機能を実行する前の状態に戻す手段を有するようにしている。

30 【0011】また、前記制御装置は、前記機能の実行が終了するまで、機能の実行による更新データを更新前データに上書きしないようにしている。

【0012】また、前記制御装置は、前記機能の実行による更新データを該制御装置内のメモリに格納するようにしている。

40 【0013】また、ネットワークを介してクライアント装置とディスク装置に接続されたサーバ装置であり、前記クライアントから前記ディスク装置における機能の実行要求を受け付けた場合、機能実行要求毎に前記ディスク装置の記憶媒体に記憶されているデータに対するアクセス範囲を制限する機能情報を生成する手段と、該機能情報を前記ディスク装置に送信する手段を有するようにしている。

【0014】また、該サーバ装置は前記クライアント装置から少なくともユーザID情報を受信し、前記機能情報を生成する手段は、該受信した少なくともユーザID情報を基にして前記アクセス範囲を制限する機能情報を生成するようにしている。

50 【0015】また、ネットワークを介してクライアント装置と接続されたディスク装置であり、データを記憶するための記憶媒体と、制御装置を備え、該制御装置は、ネットワークを介してクライアント装置から機能実行要

5

求とユーザID情報とを受信する手段と、該ユーザID情報に基づき機能実行要求毎に前記記憶媒体に記憶されているデータに対するアクセス範囲を制限する機能情報を生成し、該機能情報に基づきアクセス範囲を制限する手段を有するようにしている。

【0016】

【発明の実施の形態】《実施例1》以下図面を用いて、本実施例を説明する。図1に本発明に係わるコンピュータシステムの一実施例を示す。本実施例ではディスク装置103としてディスクドライブ118を示しているが、ディスク装置103として複数のディスクドライブ118から構成されるサブシステムでも構成可能である。図1に示すように本実施例ではクライアント101、サーバ102、ディスク装置103がネットワーク104で相互接続されている。本実施例ではサーバ102を別途設けているが、本実施例で記述しているサーバ102の同等の機能を持ったクライアント101、ディスク装置103でも実現可能である。すなわち、後述する機能サーバをクライアント101あるいはディスク装置103に実装すればよい。サーバ102はプログラムを実行するCPU105をもち、CPU105は内部バス106とメモリ制御部109を介してメモリ111へ接続される。メモリ111にはCPU105で実行されるプログラム、及び実行時に必要なデータが格納されている。本実施例ではメモリ111上に機能情報127、機能サーバ113が、ディスク装置107上に機能112が格納されている。機能112、機能サーバ113、機能情報127は実行時以外はディスク装置107に格納されており、必要に応じてCPU105が内部バス106に接続されたディスク制御部108に指示を出しディスク装置107から取り出し、使用する。機能情報127は各機能112に関する制御情報を保持する。本実施例ではディスク装置107に書き込むデータ、ディスク装置107から読み出すデータはメモリ上に格納する。サーバ102はネットワーク104にネットワーク制御部110を介して接続しており、CPU105が内部バス106を介してネットワーク制御部110を制御する。

【0017】ディスク装置103は処理を行うCPU115を持ち、CPU115で実行するプログラム、プログラム上で使用するデータはドライブI/F制御部117がCPUの指示によってドライブ118から読み出し、メモリ120上に格納する。メモリ制御部119は内部バス116を介してCPU115と接続し、CPU115からメモリ120へのアクセスに対するアクセス制御を行う。メモリ120には機能マネージャ123、機能スケジューラ125、機能122、アクセス範囲リスト124、機能情報126が格納される。機能情報126は、機能122に関する情報と、機能122のアクセス範囲を制限するアクセス範囲リスト124の情報を保持する。ディスク装置103のデータはドライブ118に格納される。ドライブ118はCPU115と内部バス116を介して接続するドライブI/F制御部117によって制御される。ドライブ118とやり取りするデータはCPU115の指示でドライブI/

6

F制御部117によってメモリ120に格納する。ディスク装置103はクライアント101、サーバ102とはネットワーク104で接続しネットワーク104はCPU115の指示でネットワーク制御部121で制御される。ネットワーク104経由でやり取りするデータはメモリ120上に格納される。

【0018】図2に本実施例に係るサーバ102のプログラム構造の一実施例を示す。サーバ102はオペレーティングシステム201内にファイルシステム202をもち、オペレーティングシステム201上に機能サーバ113を実装する。機能サーバ113は、機能情報127を用いディスク装置103で実行する機能112を管理し、機能サーバ113は必要ときにディスク装置107の機能リスト204から機能112をサーバ102上のメモリ111に読み込む。ファイルシステム202はディスク装置107に格納するデータの配置、ユーザ情報等の属性を管理する。

【0019】図3は本実施例に係る機能情報127の一実施例を示す。機能情報127はサーバ102における機能112の管理情報を格納するテーブルである。管理番号であるID301、名称302、機能の所有者、機能の実行権限およびアクセス権限を示す属性303、すでに機能をオフロード済みかを示すオフロード済みディスク装置303、機能のバージョン情報305、機能の大きさを示すサイズ306、機能の実行レベル307、機能実行時のアクセスファイルを示すアクセス対象0～n-1308で構成される。アクセス対象0～n-1 308はファイルシステム202が管理するファイルを示し、ファイルの名称、サイズ、属性の情報を含む。上記管理情報は基本的に機能の登録時にユーザにより設定される。また、アクセス対象308は機能の実行時にもユーザにより設定可能である。

【0020】図4は本実施例に係るディスク装置103のプログラム構造の一実施例を示す。ディスク装置103ではオペレーティングシステム401のドライブ制御プログラム402を用いドライブ118を制御する。ドライブ118には、サーバ102から送られた機能404が格納されており必要に応じてメモリ120上に読み込む。オペレーティングシステム401上には機能マネージャ123が実装される。機能マネージャ123は実行中の機能122のスケジューリングを行う機能スケジューラ125があり、機能スケジューラ125は実行キュー403で各機能122を管理する。実行キュー403は機能122の優先順位に従い各機能122を実行する。本実施例では優先順位1に接続している機能122が最も優先順位が高く、優先順位2、3、4と優先順位が低くなる。実行する機能122は実行キュー403につながれ、同一優先順位の機能122はリストでつながれる。機能情報00、01、02、03、04 126は機能122の管理情報であり、機能情報126と呼ぶ。各機能情報126はアクセス可能な領域を示すアクセス範囲リスト124を保持する。

【0021】図5はディスク装置103上の機能マネージャ123の動作を示す流れ図である。機能マネージャ123はディスク装置103の起動後、ドライブ118からメモリ120

上に読み込まれ実行される。まず最初に機能マネージャ123は実行キュー403、機能情報126、アクセス範囲リスト124の領域を準備する(501)。次に、要求が来るのを待ち、要求が来たときには、要求種別を解析し(503)、機能オフロードのときには機能オフロード処理(507)を実行し、機能の実行要求の場合(504)には機能実行処理(506)を行う。また上記2要求以外の場合は該当要求無しと報告する(505)。以上の処理を繰り返し実行する。

【0022】図6は本実施例に係るディスク装置103における機能122の管理情報の一実施例を示す。図6の機能情報126は機能情報127を元にサーバ102で作成し、機能の実行要求とともにディスク装置103に送付される。機能情報126は機能の管理番号であるID601、名称602、ドライブ118上の機能格納場所を示すポインタ603、機能122の大きさを示すサイズ604、実行時の優先順位を示す実行レベル605、機能122のバージョンを示すバージョン606、実行キュー403に接続したとき次の機能を指し示すための次ポインタ607(図では、図4の例にしたがって、機能00の先頭アドレスと記述している)、アクセス範囲リスト124の場所を示すポインタ608、リストの大きさを示すサイズ609、リスト中の要素数610を含む。

【0023】図7は本実施例に係るアクセス範囲リスト124の一実施例を示す。図7のアクセス範囲リスト124は機能情報127を元にサーバ102で作成され、機能の実行要求とともにディスク装置103に送付される。具体的には、機能情報127のアクセス対象308をもとに、ファイルシステム202から該アクセス対象308(ディレクトリまたはファイル)の物理アドレスと、属性(読出し(r)、書込み(w)、実行可能(x))を取得しアクセス範囲リスト124を作成する。アクセス範囲リスト124は1以上の項目を有し、各項目にはアクセス範囲の先頭アドレスを示す先頭701、大きさを示すサイズ702、データの属性を示す属性703、が含まれる。属性は読出し(r)、書込み(w)、実行可能(x)の3種がある。実行可能の場合にはプログラムとしてメモリ上から実行可能である。本実施例では物理アドレスを示したが、NASDに示されるようなオブジェクトの管理IDでも指定可能である。

【0024】まず最初に機能をディスク装置103に送信する動作について説明する。

〈機能送信〉最初に図8を用いて機能の送信を説明する。図8は機能送信の正常時の情報のやり取りを示した図である。本実施例ではクライアント101から機能を送信する手順を示しているが、機能の送信元はクライアント101以外でも実施可能である。まず最初にクライアント101は機能サーバ113に対して機能送信の要求を行う(801)。この時、送信元のユーザ情報も要求時に機能サーバ113に送信する。機能サーバ113は要求を受信し(802)、要求を解析し、ユーザの権限等をチェックし、送信先のディスク装置103上の機能マネージャ123に機能送信可能かと問い合わせる(803)。機能を送信可能であ

ば、受信可能である旨をクライアント101に通知する(804)。クライアント101は機能送信可能であると受信すると(805)、まず最初に機能に関する情報をサーバ102に送信する(806)。サーバ102は情報を受け取る(807)。この情報は機能サーバ113上の機能情報127を作成するために使用する。機能サーバ113はクライアント101から受け取った情報から機能サーバ上の機能情報127を作成した後、クライアントに送付する機能情報126を作成する(808)。

【0025】機能情報127の内容は、送信先ディスク装置103、機能の名称、サイズ、実行レベル、アクセス対象等がある。アクセス対象はファイルやディレクトリを指定する。情報を受け取った機能サーバ113はクライアント101から受け取った情報に使用ファイルに該当するディスク装置103上の物理アドレス、及び属性をファイルシステム、オペレーティングシステムから取得し機能情報126と、アクセス範囲リスト124を作成し、クライアントに送付する。ID601、名称602、実行レベル605、バージョン606は機能サーバ113の機能情報127の値を用い、ポインタ603、サイズ604は該機能が格納されているディスク装置103上の物理アドレスであり、ファイルシステム202から取得する。アクセス範囲リスト情報はアクセス範囲リスト124に関する情報を格納する。ポインタ608はアクセス範囲リスト124に対する位置情報を示すポインタ608、アクセス範囲リスト124の大きさを示すサイズ609、アクセス範囲リストの要素の数を示す要素数610を以下に示すアクセス範囲リスト124の作成後に作成時の情報を元に作成する。

【0026】図9、図10を用いアクセス範囲リスト124の作成方法を説明する。図ではオペレーティングシステムとファイルシステムに問い合わせた結果ファイルのデータがデータ0の位置とデータn-1の位置に配置されている様子を示している。図のように一つのファイルが断片化(離れ離れに配置されている)されている場合には、図9のように各連続部分データ0 902とデータn-1 903を個別にアクセス領域904とアクセス領域905でリスト中に情報906、情報907で指定する方法と、図10のようにディスク装置103の先頭データ0 1001と、終わりデータn-1 1002のを含むアクセス領域1003でリストに情報1004を指定する方法がある。図9は細かく設定するために信頼性が良い代わりに、アクセス範囲リスト124が大きく、図10ではおおざっぱに設定するために信頼性が悪くなるが、アクセス範囲リスト124のサイズが小さくなる。どちらの方式を採用するかはシステムに依存する。本方式を実装する場合にはファイルシステムが意識して連続領域にファイルを、格納することが望ましい。

【0027】機能サーバ113は情報を受け取ったのちに、クライアント101に対し、受領を通知し(809)、クライアント101は機能情報受領を受け取り(810)、次いで機能をサーバ102に送信する(811)。機能サーバ113は機能

をクライアント101から受け取ると(812)、該機能をディスク装置103に格納し機能サーバ113上の機能リストへ情報を登録する(813)。次に、サーバ102は機能情報126とアクセス範囲リスト、機能を合わせ、この合わせた情報をクライアント101から指定されたディスク装置103へ送信する(814)。ディスク装置103はこの情報を機能サーバ113から受け取ると(815)、機能リストに登録し(816)、送信終了と格納結果を機能サーバ113へ送信する(817)。機能サーバ113は、結果を受け取り(818)、クライアント101に対しても同様に結果を送信し(819)、クライアント101はこれを受信する(820)。

【0028】図11を用いてクライアント101の動作を詳細に示す。最初にクライアント101はクライアント101を操作するユーザが機能の発行要求を機能サーバ113に対して送信する(1101)。機能サーバ113から受信報告を受け(1102)、次に、機能送信時に必要な、送信先ディスク装置103、機能の名称、バージョン、サイズ、実行レベル、アクセス対象等を機能サーバ113に送信する(1103)。次に、受信報告を受け取り、エラーであれば機能送信エラーであるのでエラー処理を行い終了する(1110)。正常であれば処理を続ける。受信報告(1104、1105)を解析し、機能が機能サーバ113上に無い場合には機能を送信する(1106~1109)、機能サーバ113上にすでにある場合には送信せずにサーバ102からの報告を待ち、ディスク装置103に正常に機能が送信されたのち、機能送信が正常であればそのまま処理を終了し、エラーであればエラー処理(1110)を行い処理を終える。

【0029】図12を用いて機能サーバ113の動作を詳細に示す。最初に機能サーバ113はクライアント101から機能の発行を受け付ける(1201)。機能が送信可能か調べ(1202)、送信可であれば受け付けた旨をクライアント101に送信(1203)し、処理を続け、エラーが発生していればエラー発生をクライアント101に通知し、エラー発生処理を実行し処理を終える(1221)。次に機能情報127を受信する(1204、1205)。この時、まず最初に送信された機能情報127をチェックする。同一バージョンの当該機能が既に機能サーバ113に無い場合には機能をクライアント101から受信する(1207~1210)。機能がある場合にはディスク装置に対する機能送信処理(1211~1220)を実行する。

【0030】機能をクライアントから受信する処理(1207~1210)について説明する。該機能が機能サーバ113上に無い場合には、機能をクライアント101から受信する(1207、1208)。正常に受信した場合には受信した旨をクライアント101に伝え(1210)、エラーが発生した場合にはエラーを報告し処理を終了する(1221)。次に、受信した機能をメモリ上に格納する。次に機能をディスク装置107に格納する。機能サーバ113はサーバ102上のファイルシステム202を用い格納する。この時のファイル名は機能サーバ113がシステム上で一意に決まるように決める。

また、機能のIDもシステム上で一意に決まるよう決定する。本要求をクライアント101から送信したユーザを所有者とし、機能を誰が実行・参照・削除可能かという属性を作成し、さらにクライアント101から送信された名称、バージョン、サイズ、実行レベル、アクセス対象を追加し該機能の機能情報127を作成する。作成した機能は機能リスト204に格納し、保存する。

【0031】次にディスク装置103の機能マネージャ123に対する機能送信について説明する(1211~1220)。まず、ディスク装置103に送信済みか調べる(1211)。既にディスク装置103に送信済みの場合は送信終了通知をクライアント101に送信し処理を終える(1220)。機能が未送信の場合には以後ディスク装置103に対する機能送信処理を行う。まず最初にディスク装置103へ機能の送信要求を発行し(1212)する。次に、機能サーバ113からディスク装置103へ機能情報126と機能の送信を行う。機能情報126はアクセス範囲リスト124の情報を保持する。まず、機能送信の要求を発行し(1212)、次に機能情報126の送信を行う(1214)。機能情報126は、機能サーバ113がディスク装置103の機能マネージャ123へ送信する(1214)。正常に機能情報126が送信されたときには機能の送信(1217)を実行する。機能情報126の送信がエラーとなった場合にはエラーをクライアント101に報告し処理を終える(1216、1221)。機能情報126を正常に送信した後、機能サーバ113は機能マネージャ123に対し機能を送信する。機能を正常に送信した場合には、クライアントに正常送信の報告を行い(1220)、機能の送信が正常であった場合には機能情報126の送信済みディスク装置103に該ディスク装置103に登録したのち、クライアント101に送信処理正常終了を通知し処理を終える。機能の送信時にエラーが発生した場合にはこの時機能サーバ113上の機能と機能情報126はディスク装置103に対する送信が未送信の状態で保持する。

【0032】図13を用いて機能マネージャ123の動作を詳細に示す。最初に機能サーバ113から機能マネージャ123へ機能送信の要求が送信される(1301)。機能マネージャ123は機能を受け付けられない場合には、受付不可の旨を機能サーバ113に通知し処理を終える(1302、1311)。機能を受け付けられるときには、機能サーバ113に対して機能受付可能の旨を通知し以後処理を続ける(1302、1303)。次に、機能マネージャ123は機能サーバ113から機能情報126が送信されてくるのを待つ(1304)。機能マネージャ123は機能情報126が送信されてきたら(1305)、まず最初に送信された機能情報126を元にディスク装置103上の機能マネージャ123の機能情報126へ変換する。次に、機能マネージャ123は機能サーバ113から機能が送信されてくるのを待つ(1306)。機能が送られてきたら、機能情報126に記された機能のサイズの領域を確保しドライブ118に書き込む(1307)。この時の先頭アドレスを機能情報126へ追加する。機能が正常に格納された

11

ら、機能情報126をドライブ118の任意の領域に格納し、機能正常受信の旨を機能サーバ113に通知し(1308、1309)最後に機能を機能リスト405に登録し(1310)処理を終える。

【0033】〈機能実行〉まず最初に図14を用い処理の大まかな流れを説明する。クライアント101のユーザが今から実行したい機能と機能を実行するディスク装置103を機能サーバ113に通知する(1401)。機能サーバ113はクライアント101からの要求を受け取ると(1402)、要求内容を解析し、要求したユーザ権限の有無等を調べ(1403)、実行可能であるときには、実行可能と通知する(1404)。クライアント101は実行可能と受け取ると(1405)、機能実行に必要なパラメータを機能サーバ113に送信する(1406)。ここで、パラメータは、使用するファイル名及び機能実行時に必要なデータである。機能サーバ113はパラメータを受信し(1407)、該機能に対し関連付けられている機能情報126を機能リスト204から取り出し、ディスク装置103の機能マネージャ123に送信する機能情報126とアクセス範囲リスト124を作成する(1408)。この時、既にアクセス範囲リスト124に領域が指定されており、クライアント101から更にファイル、ディレクトリを指定された場合にはアクセス範囲リスト124に領域を追加する。機能サーバ113は機能の実行要求とともに作成した情報を機能マネージャ123に送信する(1409)。機能マネージャ123は機能の実行要求を受け取ると(1410)、機能受付の報告を機能サーバ113に通知する(1411)。次に機能マネージャ123は受け付けた機能を実行し(1413)、機能が正常に終了したときには、終了と機能の実行結果を機能サーバ113に通知し(1414)、機能サーバ113は該通知を受け(1415)、解析したのちクライアント101に対して終了と機能の実行結果を通知する(1416)。クライアント101は終了と機能の実行結果を受け取り(1417)、ユーザに終了を報告し処理を終える。

【0034】次に図15を用い機能の実行要求の流れを説明する。クライアント101のユーザがある機能を実行するときには、最初に機能サーバ113に機能実行の要求を送信する(1501)。機能実行の要求とともに機能を実行するディスク装置103とユーザを通知する。ここで、機能を実行するディスク装置103としたが、サーバ102上のファイルシステム202のファイルからディスク装置103が一意に決定可能であるため、サーバ102上のファイルシステムのファイルを指定しても実施可能である。クライアント101は、機能の実行要求を発行した後、機能サーバ113から機能の受付の報告を受け取る(1502)。機能がサーバに登録されて無い場合には機能の送信処理を行い(1504)、送信が正常に行われたときには処理を続け、エラー発生時にはエラーを通知し処理を終える(1512)。次に、報告の内容が実行不可能である場合には、機能は実行不可能なため、エラー処理を行い処理を終了する(1506、1512)。機能を実行可能な場合には次に機能実行時に

12

必要なパラメータを送信する(1507)。ここで、パラメータは、使用するファイル、ディレクトリ及び機能実行時に必要なデータである。パラメータを送信した後、パラメータの受信結果を受け取り(1508)、エラーの場合はエラー処理を行い処理を終了する。正常の場合には処理を続ける。機能の実行終了を待ち(1510)、クライアント101は機能終了と機能の実行結果を受け付け(1511)、エラー終了の場合はエラー処理を行いユーザにエラーが発生した旨を通知し処理を終了する(1512)。正常に処理が行われたときにはユーザに正常処理の旨と機能実行結果を通知し処理を終える(1513)。

【0035】次に図16を用い機能サーバ113の機能実行時の処理について説明する。まず最初に機能サーバ113はクライアント101から機能の実行要求を受け取る(1601)。この時、クライアント101は要求とともに機能を実行するディスク装置103と要求元のユーザを指定する。機能サーバ113は機能リスト204から要求機能を検索する。要求機能が機能リスト204に無いか、又は機能情報126を調べ該機能が該ディスク装置103に送信済みでない場合には(1602)、クライアント101にある機能を機能サーバ113経由で機能マネージャ123に対し機能を機能サーバ113に送信する(1603)。機能送信が正常に終了した場合には処理を継続する。機能が正常に送信できなかったときには機能送信を失敗した旨をクライアント101に通知し処理を終える(1604、1605)。機能がディスク装置103に送信されているときには機能リスト204から該機能の機能情報126を取り出し、該機能に対して要求元ユーザが機能を実行する権限が在るか調べ(1606)、機能の実行権限が無い場合には実行権限の無い旨をクライアント101に伝え処理を終える(1607、1608)。機能の実行権限がある場合には、実行可能の旨をクライアント101に伝える(1607、1609)。

【0036】次に、クライアント101から機能実行時に必要なパラメータを受け取り(1610)、アクセス範囲に関するパラメータは機能情報126のアクセス対象に追加する。次にアクセス対象の変換を行う。アクセス対象はサーバ102上のファイルシステム202及びオペレーティングシステム201からファイルまたは、ディレクトリの物理アドレスを取得し、ファイルの物理位置を求め、アクセス範囲リスト124を作成する。この時要求元ユーザが指定されたファイル、ディレクトリに対してのアクセス権限が在るかを調べる(1612)。アクセス権限が無い場合にはクライアント101に処理エラーを通知し、処理を終える(1608)。すべてのアクセス対象に対しアクセスが可能である場合には処理を続ける。

【0037】次に機能サーバ113は機能の実行要求とともに機能マネージャ123に対し、アクセス範囲以外のパラメータ、アクセス範囲リスト124を送信する(1613)。機能実行要求の受付報告(1614)がエラーのときにはクライアント101にエラーを通知し処理を終える(1618)。正

常のときには機能の実行終了の報告を待つ(1615)。機能の実行終了後、機能サーバ113は機能マネージャ123から機能終了の報告を受け取った後に、報告内容を調べエラーであればエラーをクライアント101に通知し処理を終える(1616、1618)。報告が正常であれば、正常終了の旨と機能の実行結果をクライアント101に通知し処理を終える(1616、1617)。上記に示したように機能が機能マネージャ123に対して送信されていない場合には、機能マネージャ123に対する機能の実行要求の前に機能の送信処理の手順に従い機能の送信処理が行われる。

【0038】以下に機能が既に機能マネージャに送信済みのときに機能を実行する処理手順について説明する。図17を用いて機能実行時の機能マネージャ123の動作の説明を行う。機能マネージャ123は、機能の実行要求を機能サーバ113から受け取る、実行要求とともに実行時に必要なパラメータ及びアクセス範囲リスト124を受け取る(1701)。アクセス範囲リスト124にはアクセス可能な領域と、領域のサイズ、属性が格納されている。機能サーバ113から送信された情報を機能情報126、アクセス範囲リスト124に登録した後、該機能を実行キュー403に挿入する。機能情報126の実行レベルから優先順位を決め、機能を挿入する。機能マネージャ123の機能スケジューラ125は優先順位の高い機能から順番に実行する。実行中の機能が実行終了するか、一定の時間CPUで実行するか、I/O等のCPU使用せずに待ちが発生した場合には、実行中の機能を実行キュー403から取り出し、次に優先度の高い機能を実行する。一定時間CPUで実行された機能は再び実行キュー403の当該優先順位の最後に挿入される。I/O等の待ちになった機能は待ち状態が解除されたのち、再び実行キュー403に挿入される。上記に示すスケジューリングで機能を実行していく。

【0039】機能実行時にドライブ118へのアクセスが発生すると(1702)、機能マネージャ123はアクセス先のアドレスがアクセス範囲リスト124の範囲内か調べる(1703)。アクセスリストの範囲外に対するアクセスのときにはエラーであり、許可範囲外アクセスエラーを機能マネージャ123に通知し処理を終える(1709)。アクセス範囲内のときにはアクセス種別を調べる(1704)。リードのみ可のときにアクセス範囲に対するライトなどアクセスするデータの属性がアクセス内容と異なる場合には先ほどと同様に許可範囲外アクセスエラーを機能マネージャ123に通知し処理を終える(1709)。アクセス種別が指定範囲内にあれば、処理を実行する(1705)。上記の処理を機能の実行が終了するまで続け(1707)、機能が正常に終了したときには機能マネージャ123は機能サーバ113に対し正常終了と機能の実行結果を通知し処理を終える(1708)。

【0040】《実施例2》以下、図面を用いて、本実施例2を説明する。本実施例の特長は、機能の実行時にエラーが発生した場合に、機能の更新途中の状態を残さ

ず、機能の実行前に戻せる事(ロールバック)である。図18に本発明に係わるコンピュータシステムの一実施例を示す。本実施例ではディスク装置103としてディスクドライブ118を示しているが、ディスク装置103として複数のディスクドライブ118から構成されるサブシステムでも構成可能である。図18に示すように本実施例ではクライアント101、サーバ102、ディスク装置103がネットワーク104で相互接続されている。本実施例ではサーバ102を別途設けているが、本実施例で記述しているサーバ102の同等の機能を持ったクライアント101、ディスク装置103でも実施可能である。サーバ102はプログラムを実行するCPU105を持ち、CPU105は内部バス106とメモリ制御部109を介してメモリ111へ接続される。メモリ111にはCPU105で実行されるプログラム、及び実行時に必要なデータが格納されている。本実施例ではメモリ111上に機能情報127、機能サーバ113が、ディスク装置107上に機能112が格納されている。機能112、機能サーバ113、機能情報127は実行時以外はディスク装置107に格納されており、必要に応じてCPU105が内部バス106に接続されたディスク制御部108に指示を出しディスク装置107から取り出し、使用する。機能情報127は各機能112に関する制御情報を保持する。本実施例ではディスク装置107に書き込むデータ、ディスク装置107から読み出すデータはすべてメモリ111上に格納する。サーバ102はネットワーク104にネットワーク制御部110を介して接続しており、CPUが内部バスを介してネットワーク制御部110を制御する。

【0041】ディスク装置103は処理を行うCPU115を持ち、CPU115で実行するプログラム、プログラム上で使用するデータはドライブ118からドライブI/F制御部117がCPUの指示によってドライブ118から読み出し、メモリ120上に格納する。メモリ120には機能マネージャ123、機能スケジューラ125、機能、アクセス範囲リスト124、キャッシュメモリ1801、キャッシュ管理テーブル1802が格納される。メモリ制御部119は内部バス116を介してCPU115と接続し、CPU115からメモリ120へのアクセスに対してアクセス制御を行う。メモリ120には機能マネージャ123、機能スケジューラ125、機能122、アクセス範囲リスト124、機能情報126が格納される。機能情報126は、機能122に関する情報と、機能122のアクセス範囲を制限するアクセス範囲リスト124の情報を保持する。ディスク装置103のデータはドライブ118に格納される。ドライブ118はCPUと内部バスを介して接続するドライブI/F制御部117によって制御される。ドライブ118とやり取りするデータはCPU115の指示でドライブI/F制御部117によってメモリ120上に格納する。ディスク装置103はクライアント101、サーバ102とはネットワーク104で接続し、ネットワーク104はCPU115の指示でネットワーク制御部121で制御される。ネットワーク104経由でやり取りするデータはメモリ120上に格納される。

【0042】図19は本実施例に係るディスク装置103のプログラム構造の一実施例を示す。ディスク装置103ではオペレーティングシステム401内のドライブ制御プログラム402でドライブ118を制御する。ドライブ118には、サーバ102から送られた機能が格納されており必要に応じてメモリ120上に読み込む。オペレーティングシステム401上には機能マネージャ123が実装される。機能マネージャ123は実行中の機能のスケジューリングを行う機能スケジューラ125があり、機能スケジューラ125は実行キュー403で各機能を管理する。実行キュー403は機能の優先順位に従い各機能を実行する。本実施例では優先順位1に接続している機能が最も優先順位が高く、優先順位2、3、4と優先順位が低くなる。実行する機能は実行キュー403につながれ、同一優先順位のキューはリストでつながれる。機能情報00、01、02、03、04 126は機能122の管理情報であり、機能情報126と呼ぶ。各機能情報126はアクセス可能な領域を示すアクセス範囲リスト124を保持する。更に、ドライブ118に対するアクセスデータを管理するためのキャッシュメモリ1801とキャッシュ管理テーブル1802を機能マネージャ123内に実装し、ドライブ118に対する機能のロールバックを可能とする。本実施例ではキャッシュメモリ1801をメモリ120に実装したが、更新データが大量になったときにはメモリ120に格納しきれなくなる恐れがあるため、ドライブ118に更新データ用のキャッシュメモリを用意する事も可能である。機能サーバ113及び機能サーバ113の機能情報126は実施例1と同様である。

【0043】図20は本実施例に係る機能情報126の一実施例を示す。機能情報126は機能の管理番号であるID301、名称302、機能の所有者、機能の実行権限およびアクセス権限を示す属性303、すでに機能をオフロード済みかを示すオフロード済みディスク装置303、ファイルシステム上の機能名称であるファイル名304、機能のバージョン情報305、機能のロールバックの指示を示すロールバックフラグ2001、機能の大きさを示すサイズ306、機能の実行レベル307、機能実行時のアクセスファイルを示すアクセス対象0～n-1 308で構成される。アクセス対象0～n-1 308はファイルシステム202が管理するファイルを示し、ファイルの名称、サイズ、属性の情報を含む。アクセス範囲リスト124は実施例1と同様である。機能の送信処理は実施例1と同様に行う。

【0044】次に、機能の実行の流れを説明する。

〈機能実行〉次に図21を用い機能の実行要求の流れを説明する。クライアント101のユーザがある機能を実行するときには、最初に機能サーバ113に機能実行の要求を通信する(2101)。機能実行の要求とともに機能を実行するディスク装置103とユーザを通知する。ここで、機能を実行するディスク装置103としたが、サーバ102上のファイルシステム202のファイルからディスク装置103が一意に決定可能であるため、サーバ102上のファイルシ

ステム202のファイルを指定しても実施可能である。クライアント101は、機能の実行要求を発行した後、機能サーバ113から機能の受付の報告を受け取る(2105)。報告の内容が機能が実行不可能である場合には、機能は実行不可能なため、エラー処理を行い処理を終了する。機能を実行可能な場合には次に機能実行時に必要なパラメータを送信する(2106)。

【0045】ここで、パラメータは、使用するファイル、ディレクトリ及び機能実行時に必要なデータおよび、ロールバックの指示である。実施例1と異なるのはパラメータにロールバック指示する点にある。ロールバック指示をしない場合には機能の途中でエラーが発生しても機能の実行途中のまま実行途中のデータが残る。ロールバックの指示がある場合には機能の実行途中にエラーが発生したときに機能の実行前の状態にデータが戻る。

【0046】パラメータを送信した後、パラメータの受信結果を受け取りエラーの場合はエラー処理を行い処理を終了する。正常の場合には処理を続ける。機能の実行終了を待ち、クライアント101は機能終了と機能の実行結果を受け付け(2117)、エラー終了の場合はエラー処理を行いユーザにエラーが発生した旨を通知し処理を終了する。正常に処理が行われたときにはユーザに正常処理の旨と機能実行結果を通知し処理を終える。なお、機能サーバ、機能マネージャの機能実行についての説明は、図14についての説明と同様なので、省略する。

【0047】次に、図22を用い機能の実行要求の流れを説明する。クライアント101のユーザがある機能を実行するときには、最初に機能サーバ113に機能実行の要求を送信する(2201)。機能実行の要求とともに機能を実行するディスク装置103とユーザを通知する。ここで、機能を実行するディスク装置103としたが、サーバ102上のファイルシステム202のファイルからディスク装置103が一意に決定可能であるため、サーバ102上のファイルシステムのファイルを指定しても実施可能である。クライアント101は、機能の実行要求を発行した後、機能サーバ113から機能の受付の報告を受け取る(2202)。機能がサーバに登録されて無い場合には機能の送信処理を行い(2204)、送信が正常に行われたときには処理を続け、エラー発生時にはエラーを通知し処理を終える(2212)。

【0048】次に、報告の内容が実行不可能である場合には、機能は実行不可能なため、エラー処理を行い処理を終了する(2206、2212)。機能を実行可能な場合には次に機能実行時に必要なパラメータを送信する(2207)。ここで、パラメータは、使用するファイル、ディレクトリ及び機能実行時に必要なデータとロールバックの指示である。パラメータを送信した後、パラメータの受信結果を受け取り(2208)エラーの場合はエラー処理を行い処理を終了する。正常の場合には処理を続ける。機能の実行終了を待ち(2210)、クライアント101は機能終了と機能

の実行結果を受け付け(2209)、エラー終了の場合はエラー処理を行いユーザにエラーが発生した旨を通知し処理を終了する(2212)。正常に処理が行われたときにはユーザに正常処理の旨と機能実行結果を通知し処理を終える(2213)。

【0049】次に、図23を用い機能サーバ113の機能実行時の処理について説明する。まず最初に機能サーバ113はクライアント101から機能の実行要求を受け取る(2301)。この時、クライアント101は要求とともに機能を実行するディスク装置103と要求元のユーザを指定する。機能サーバ113は機能リスト204から要求機能を検索する。要求機能が機能リスト204に無いか、又は機能情報127を調べ該機能が該ディスク装置103に送信済みでない場合には(2302)、クライアント101にある機能を機能サーバ113経由で機能マネージャ123に対し機能を機能サーバ113に送信する(2303)。機能送信が正常に終了した場合には処理を継続する。機能が正常に送信できなかったときには機能送信を失敗した旨をクライアント101に通知し処理を終える(2304、2305)。機能がディスク装置103に送信されているときには機能リスト204から該機能の機能情報126を取り出し、該機能に対して要求元ユーザが機能を実行する権限が在るか調べ(2306)、機能の実行権限が無い場合には実行権限の旨をクライアント101に伝え処理を終える(2307、2308)。機能の実行権限がある場合には、実行可能の旨をクライアント101に伝える(2307、2309)。

【0050】次に、クライアント101からロールバックの指示を含む機能実行時に必要なパラメータを受け取り(2310)、アクセス範囲に関するパラメータは機能情報127のアクセス対象に追加する。次にアクセス対象の変換を行う。アクセス対象はサーバ102上のファイルシステム202及びオペレーティングシステム201からファイルまたは、ディレクトリの物理アドレスを取得し、ファイルの物理位置を求め、アクセス範囲リスト124を作成する。この時要求元ユーザが指定されたファイル、ディレクトリに対してのアクセス権限が在るかを調べる(2312)。アクセス権限が無い場合にはクライアント101に処理エラーを通知し、処理を終える(2308)。すべてのアクセス対象に対しアクセスが可能である場合には処理を続ける。

【0051】次に機能サーバ113は機能の実行要求とともに機能マネージャ123に対し、アクセス範囲以外のロールバック指示を含むパラメータ、アクセス範囲リスト124を送信する(2313)。機能実行要求の受付報告(2314)がエラーのときにはクライアント101にエラーを通知し処理を終える(2318)。正常のときには機能の実行終了の報告を待つ(2315)。機能の実行終了後、機能サーバ113は機能マネージャ123から機能終了の報告を受け取った後に、報告内容を調べエラーであればエラーをクライアント101に通知し処理を終える(2316、2318)。報告が正

常であれば、正常終了の旨と機能の実行結果をクライアント101に通知し処理を終える(2316、2317)。上記に示したように機能が機能マネージャ123に対して送信されていない場合には、機能マネージャ123に対する機能の実行要求の前に機能の送信処理の手順に従い機能の送信処理が行われる。

【0052】以下に機能が既に機能マネージャに送信済みのときに機能を実行する処理手順について説明する。図24を用いて機能実行時の機能マネージャ123の動作の説明を行う。機能マネージャ123は次に機能の実行要求を機能サーバ113から受け取る、実行要求とともに実行時に必要なパラメータ、ロールバック指示及びアクセス範囲リスト124を受け取る(2401)。アクセス範囲リスト124にはアクセス可能な領域と、領域のサイズ、属性が格納されている。機能サーバ113から送信された情報を機能情報126、アクセス範囲リスト124に登録した後、該機能を実行キュー403に挿入する。実行キュー403のスケジューリングは実施例1と同様に行う。実行キュー403に挿入された後の処理を図24を用いて説明する。まず最初にロールバック指示の有無を調べ(2402)、ロールバック指定があればロールバックフラグをON(2404)とし、無ければロールバックフラグをOFFとする(2403)。次に処理の実行中ドライブ118へのアクセスが無い場合はそのまま実行し、ドライブ118へのアクセスがある場合には実施例1と同様、アクセス範囲のチェックと、アクセス種別のチェックを行う(2406、2407)。該チェック時にエラーとなった場合にはロールバック処理を行う。ロールバック処理は以下のように行う。ロールバックフラグがONのときには(2412)、キャッシュメモリ1801上の未更新データを破棄(2413)した後、機能マネージャ123に対しエラーの通知を行う(2414)。ロールバックフラグがOFFのときには、キャッシュメモリ1801上の更新データをドライブ118に書き込んだ(2415)後、機能マネージャ123に対しエラーの通知を行う(2414)。アクセスチェックが正常のときには処理を実行する(2408)。処理実行の結果エラーが発生した場合には上記のロールバック処理を行う。

【0053】機能の実行が正常に終了したときにはキャッシュメモリ1801上の未更新データをドライブ118に書き込み機能の正常終了を機能マネージャ123に通知する。

【0054】次に、図25を用いて機能実行時のドライブ118に対する10の処理を説明する。機能実行時にドライブ118へのアクセスの場合には、本処理を実行する。機能マネージャ123はまず最初にドライブ118に対する処理種別を調べる(2501)。ライトの場合はキャッシュメモリ1801上にデータがある場合は、旧データに対し新しい更新をデータ上書きする。キャッシュメモリ1801上にデータが存在しない場合にはキャッシュメモリ1801上に新たに領域を確保し、確保した領域に対しデータを格納す

る(2502、2503、2504)。ライト時にキャッシュメモリ1801の内容を更新したときにはキャッシュ管理テーブル1802を更新する(2505)。ドライブ118へのアクセス種別がリードのときにデータがキャッシュメモリ1801上に存在する場合にはキャッシュメモリ1801上のデータをリードデータとし(2507)、キャッシュメモリ1801上に存在しない場合にはドライブ118上のデータをリードデータとし(2508)処理を続ける。図26は、ドライブからリードする場合を図示し、図27は、キャッシュへライトする場合を図示し、図28は、キャッシュからリードする場合を図示し、図29は、ドライブへライトする場合を図示したものである。

【0055】

【発明の効果】本発明によれば、ディスク装置における機能実行時のデータアクセスに対してデータアクセスの制限を効果的に行うことができる。

【0056】また、種々のOSにおいても同様にデータアクセスの制限ができ、制限情報の実装も実装も容易に行うことができる。

【図面の簡単な説明】

【図1】本発明の実施例1における全体の構成を示す図である。

【図2】サーバの構成を示す図である。

【図3】サーバの機能の機能情報を示す図である。

【図4】ディスク装置の構成を示す図である。

【図5】機能マネージャの処理のフローチャートを示す図である。

【図6】ディスク装置の機能の機能情報を示す図である。

【図7】ディスク装置の機能のアクセス範囲リストを示す図である。

【図8】機能送信処理の全体の正常時の処理の流れを示す図である。

【図9】アクセス範囲の指定方法の一例を示す図である。

【図10】アクセス範囲の指定方法の一例を示す図である。

【図11】クライアントにおける機能送信処理のフローチャートを示す図である。

【図12】機能サーバにおける機能送信処理のフローチャートを示す図である。

【図13】機能マネージャにおける機能送信処理のフローチャートを示す図である。

【図14】機能実行処理の全体の正常時の処理の流れを示す図である。

【図15】クライアントにおける機能実行処理のフローチャートを示す図である。

【図16】機能サーバにおける機能実行処理のフローチャートを示す図である。

【図17】機能マネージャにおける機能実行処理のフ

ーチャートを示す図である。

【図18】本発明の実施例2における全体の構成を示す図である。

【図19】ディスク装置の構成を示す図である。

【図20】ディスク装置上の機能の機能情報を示す図である。

【図21】機能実行処理の全体の正常時の処理の流れを示す図である。

【図22】クライアントにおける機能実行処理のフローチャートを示す図である。

【図23】機能サーバにおける機能実行処理のフローチャートを示す図である。

【図24】機能マネージャにおける機能実行処理のフローチャートを示す図である。

【図25】機能マネージャにおけるドライブに対するIO処理のフローチャートを示す図である。

【図26】ドライブからのリードの動作を示す図である。

【図27】キャッシュへのライトの動作を示す図である。

【図28】キャッシュからのリードの動作を示す図である。

【図29】ドライブへのライトの動作を示す図である。

【符号の説明】

101 クライアント

102 サーバ

103 ディスク装置

104 ネットワーク

105、115 CPU

106、116 内部バス

107 ディスク装置

108 ディスク制御部

109 メモリ制御部

110、121 ネットワーク制御部

111、120 メモリ

112、122、404 機能

113 機能サーバ

114 ディスク制御装置

117 ドライブI/F制御部

118 ドライブ

123 機能マネージャ

124 アクセス範囲リスト

125 機能スケジューラ

126、127 機能情報

1801 キャッシュメモリ

1802 キャッシュ管理テーブル

201、401 オペレーティングシステム

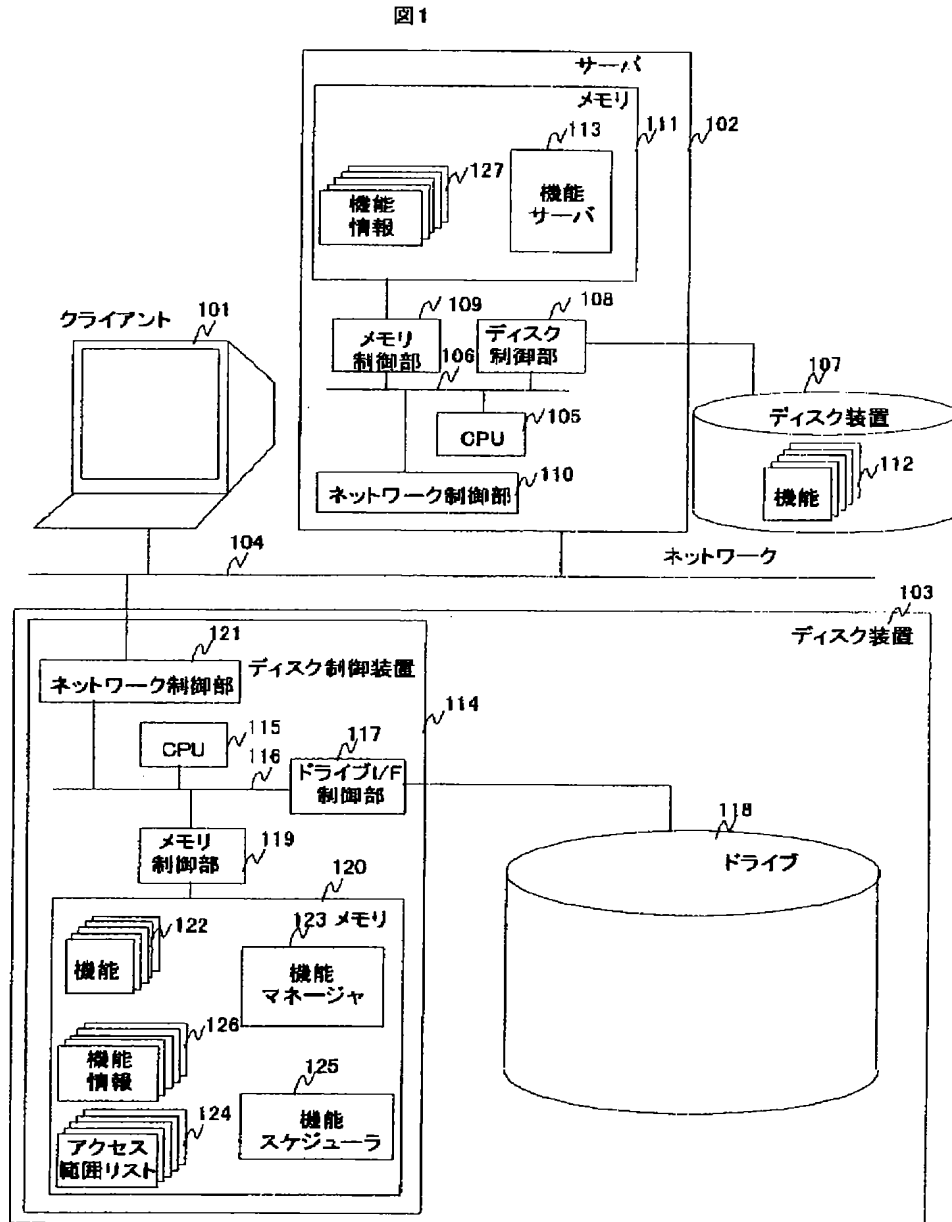
202 ファイルシステム

204、405 機能リスト

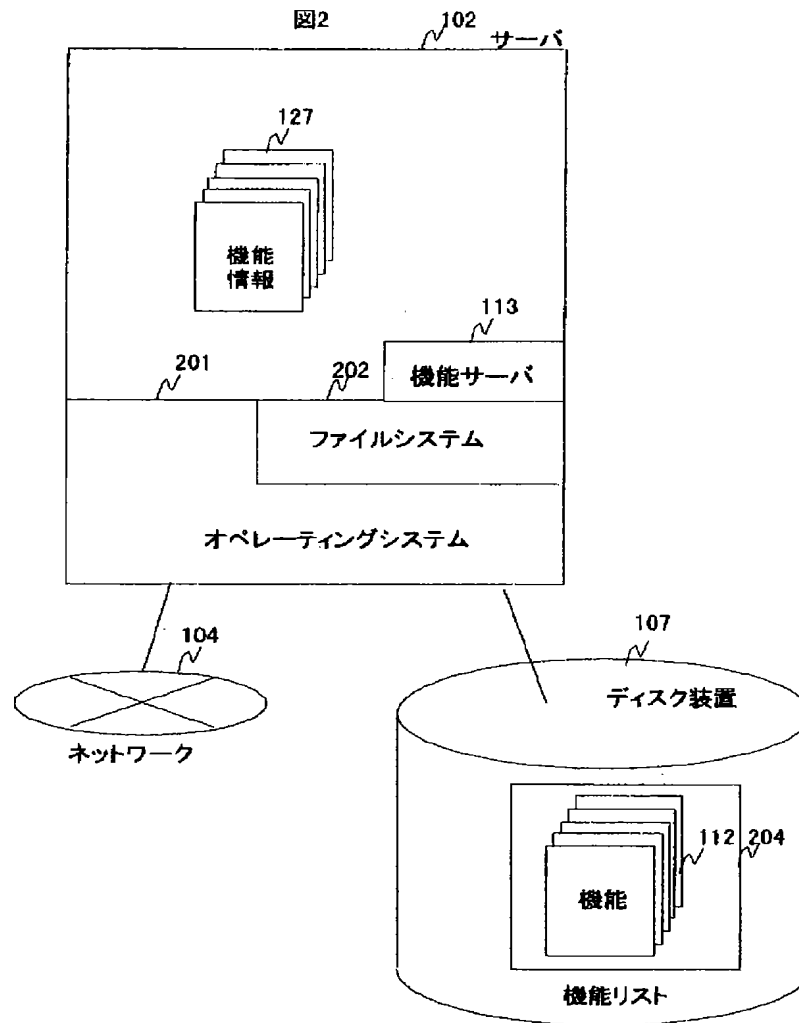
402 ドライブ制御プログラム

403 実行キュー

【図1】



【図2】



【図6】

図6 機能情報 126

ID	01	601
名称	機能01	602
ポインタ	0x0000f000	603
サイズ	0x00000100	604
実行レベル	優先度1	605
バージョン	1.006	606
次ポインタ	機能00の先頭アドレス	607
アクセス範囲 リスト情報	ポインタ	0x0000 1000 608
	サイズ	0x0000 0010 609
	要素数	n 610

【図3】

図3

127 機能情報	
ID	301
名称	302
ファイルシステム属性情報	303
オフロード済みディスク装置	304
バージョン	305
サイズ	306
実行レベル	307
アクセス対象0	308
アクセス対象1	308
⋮	
アクセス対象n-1	308

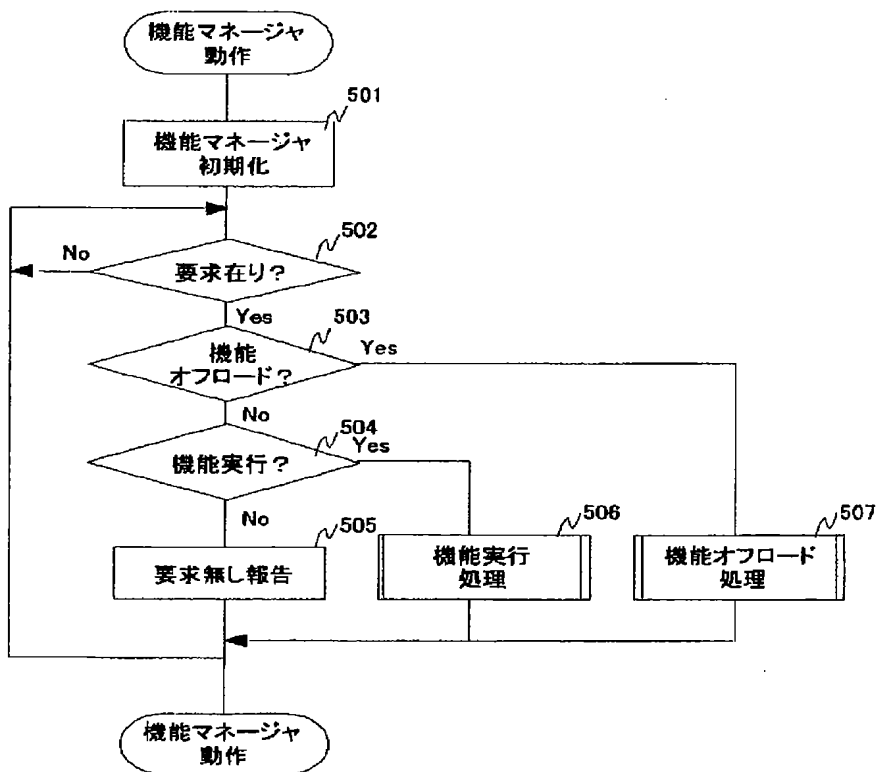
【図7】

図7
アクセス範囲リスト 124

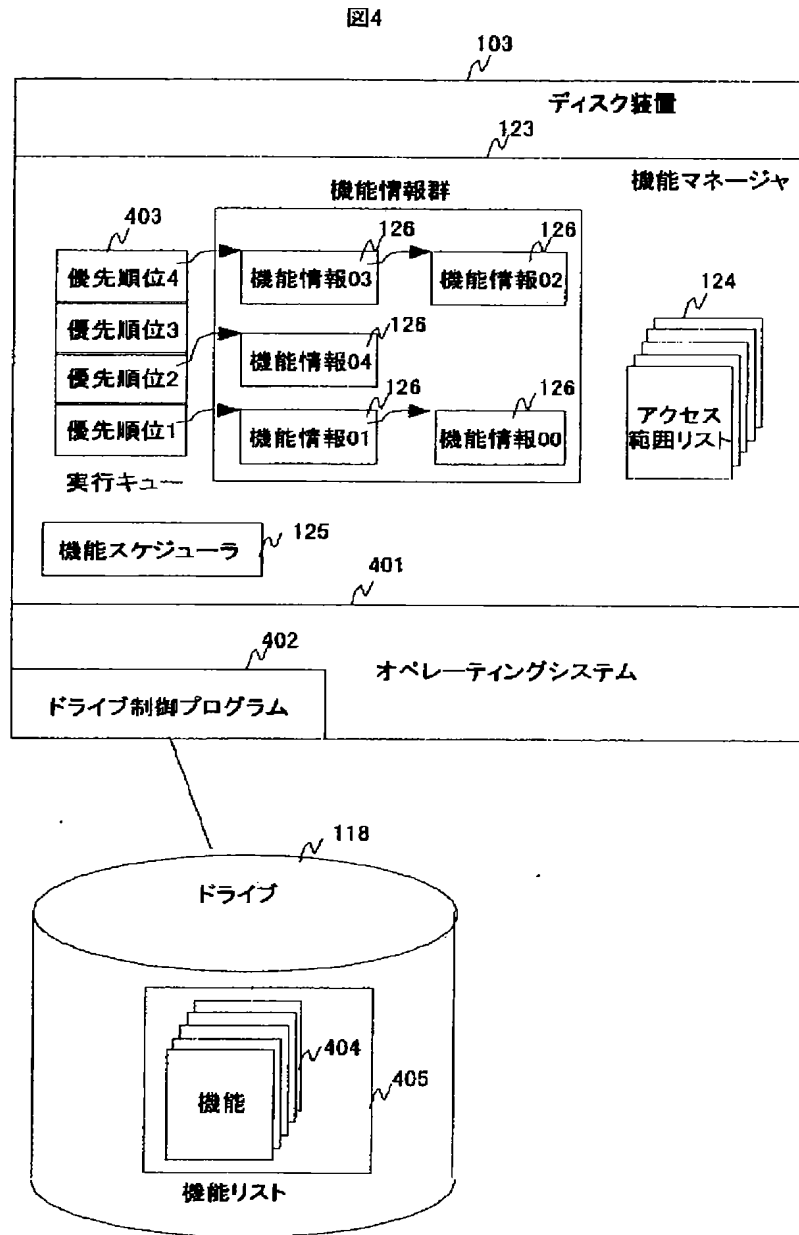
アクセス範囲 0	先頭	0x0000 0500	701
	サイズ	0x0000 0200	702
	属性	r - -	703
アクセス範囲 1	先頭	0x0000 0300	701
	サイズ	0x0000 0051	702
	属性	r w -	703
⋮	⋮	⋮	
アクセス範囲 n-1	先頭	0x0000 1100	701
	サイズ	0x0000 0020	702
	属性	r w x	703

【図5】

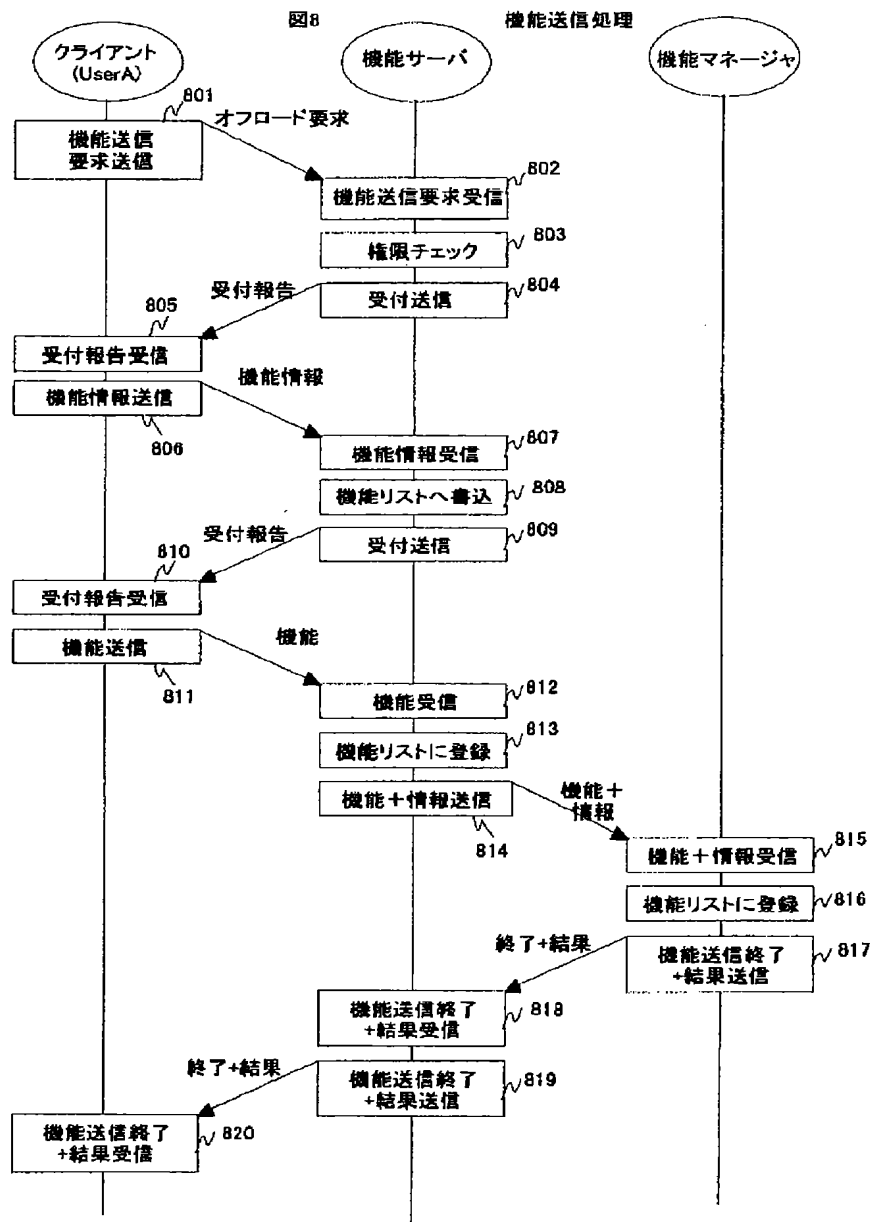
図5



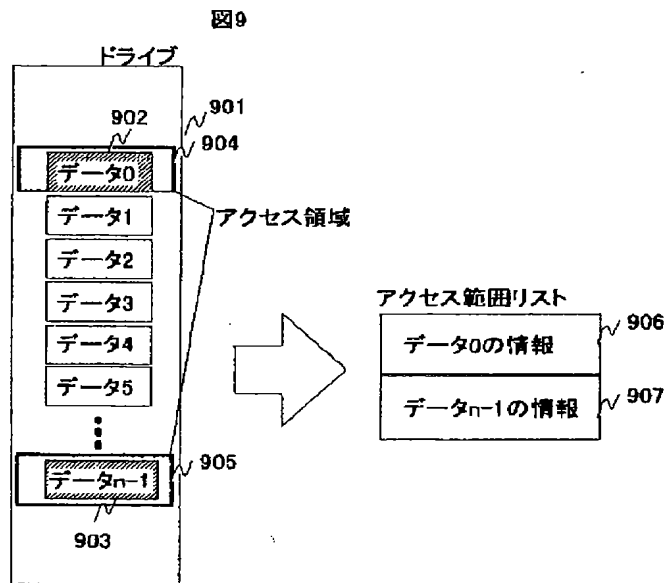
【図4】



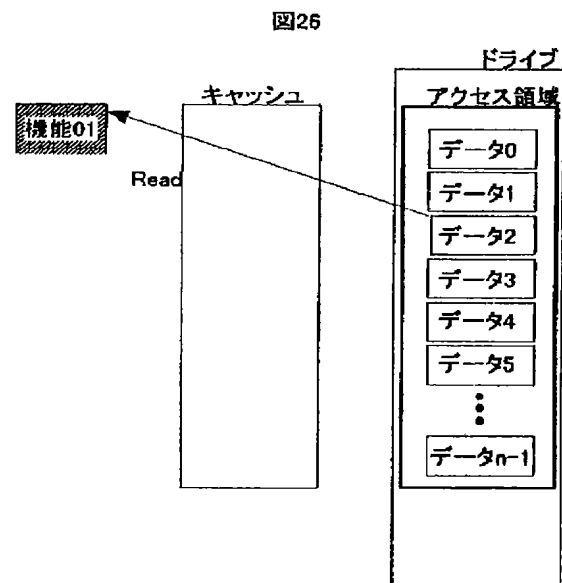
【図8】



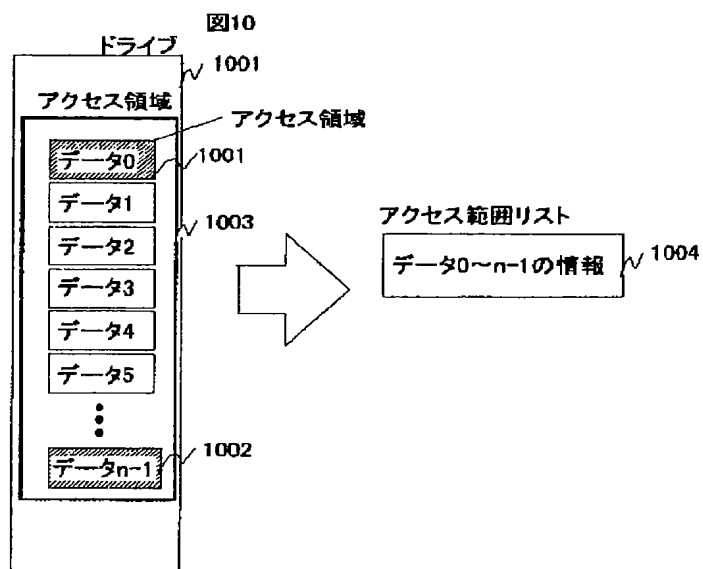
【図9】



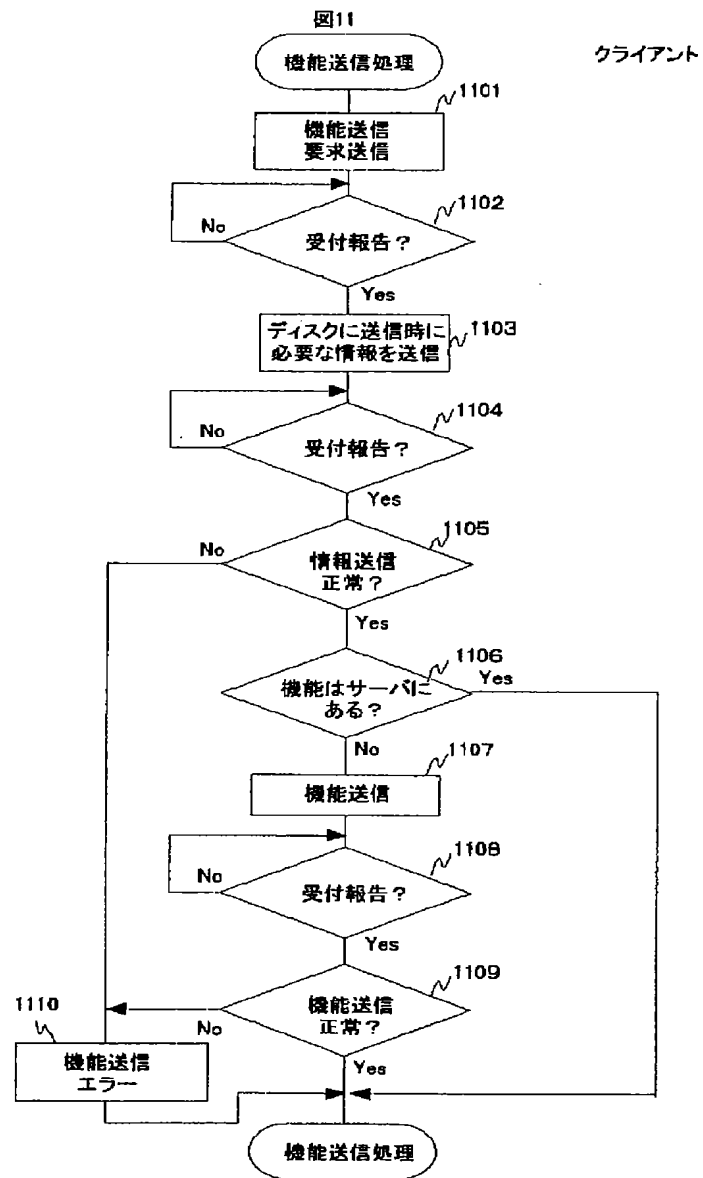
【図26】



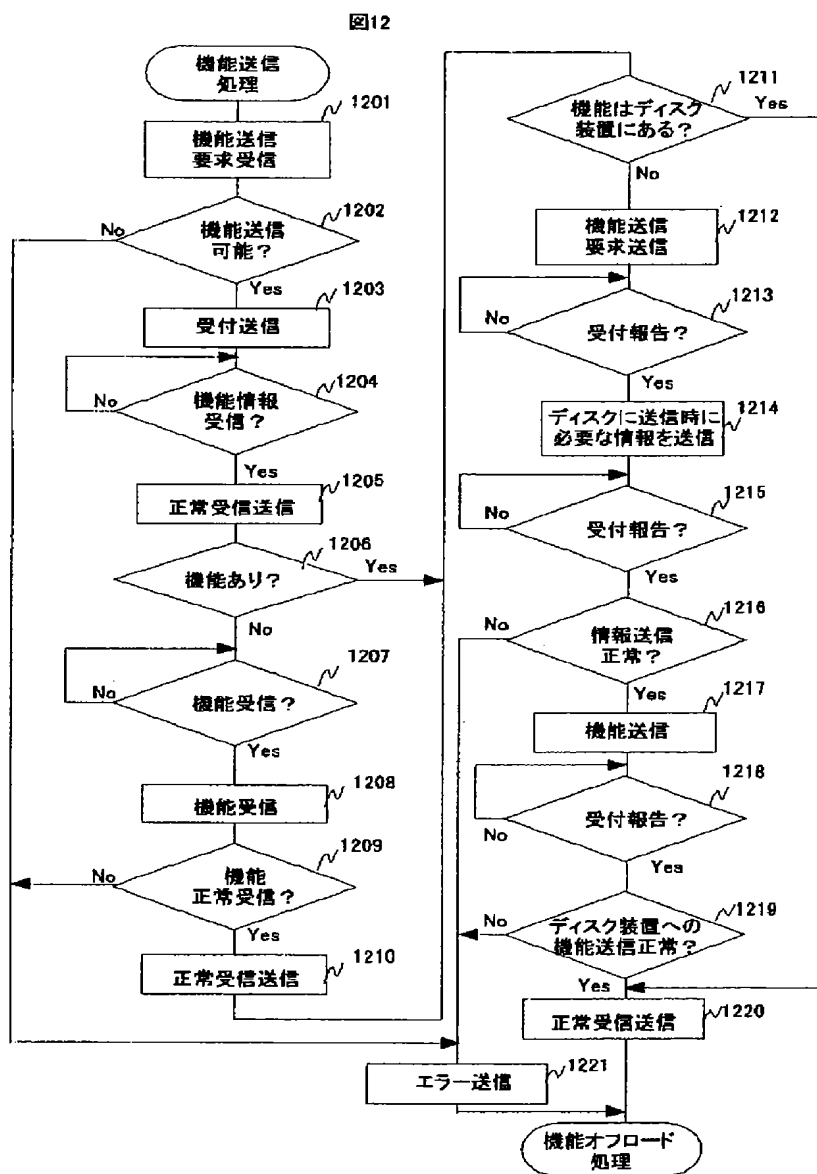
【図10】



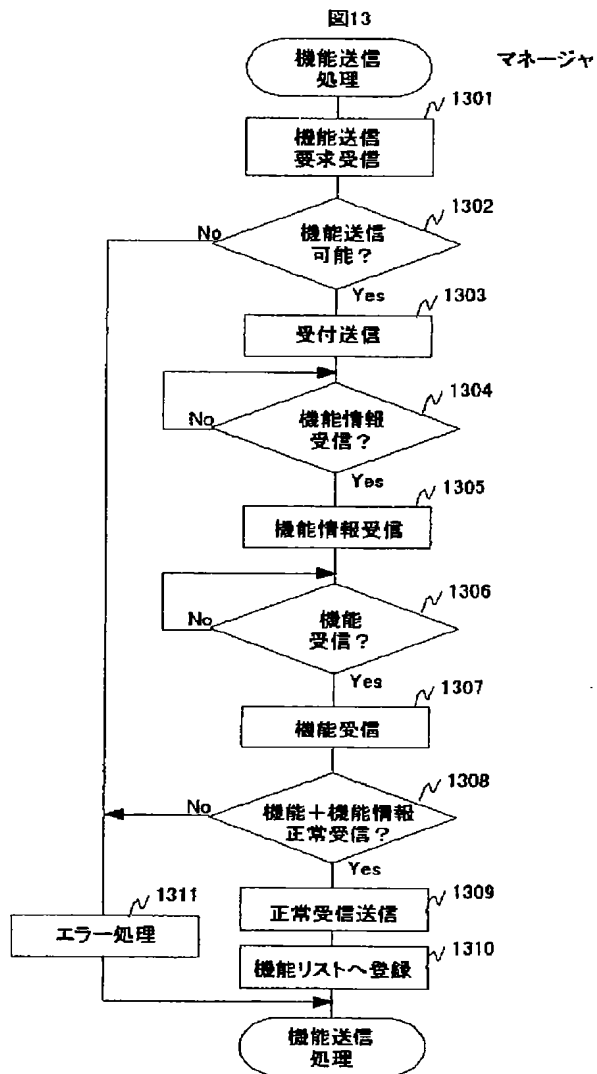
【図11】



【図12】



【図13】

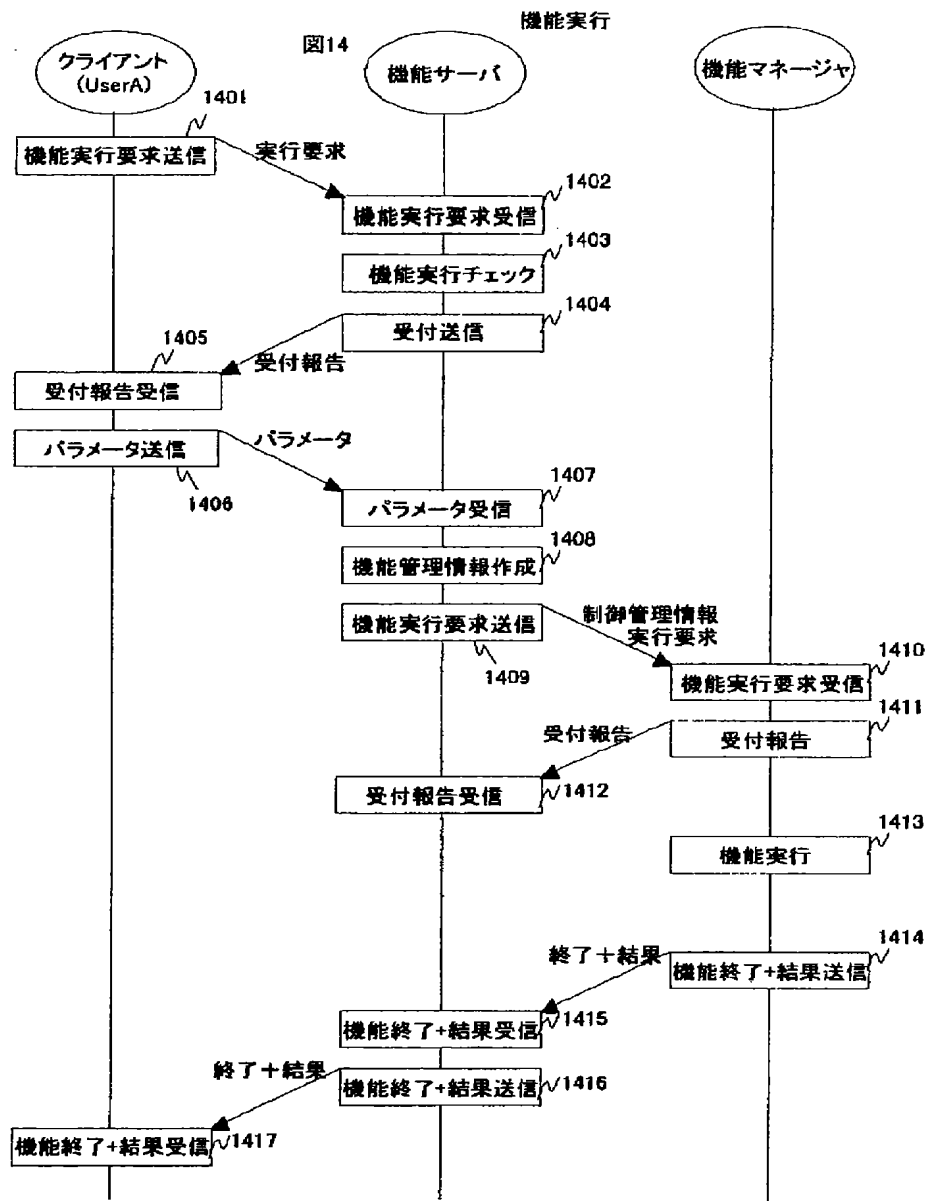


【図20】

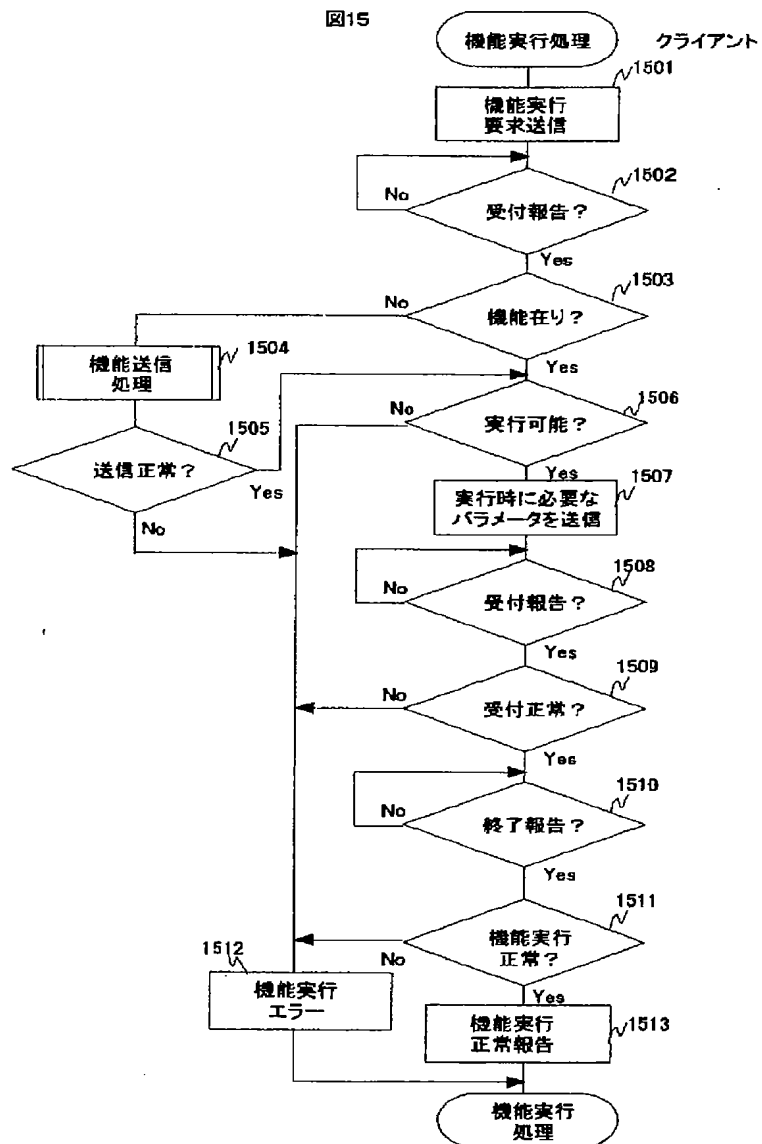
図20 125 機能情報

ID	01	601
名称	機能01	602
ポインタ	0x0000f000	603
サイズ	0x00000100	604
実行レベル	優先度1	605
バージョン	1.006	606
ロールバックフラグ	ロールバックON	2001
次ポインタ	機能00の先頭アドレス	
アクセス範囲 リスト情報	ポインタ	0x0000 1000
	サイズ	0x0000 0010
	要素数	n

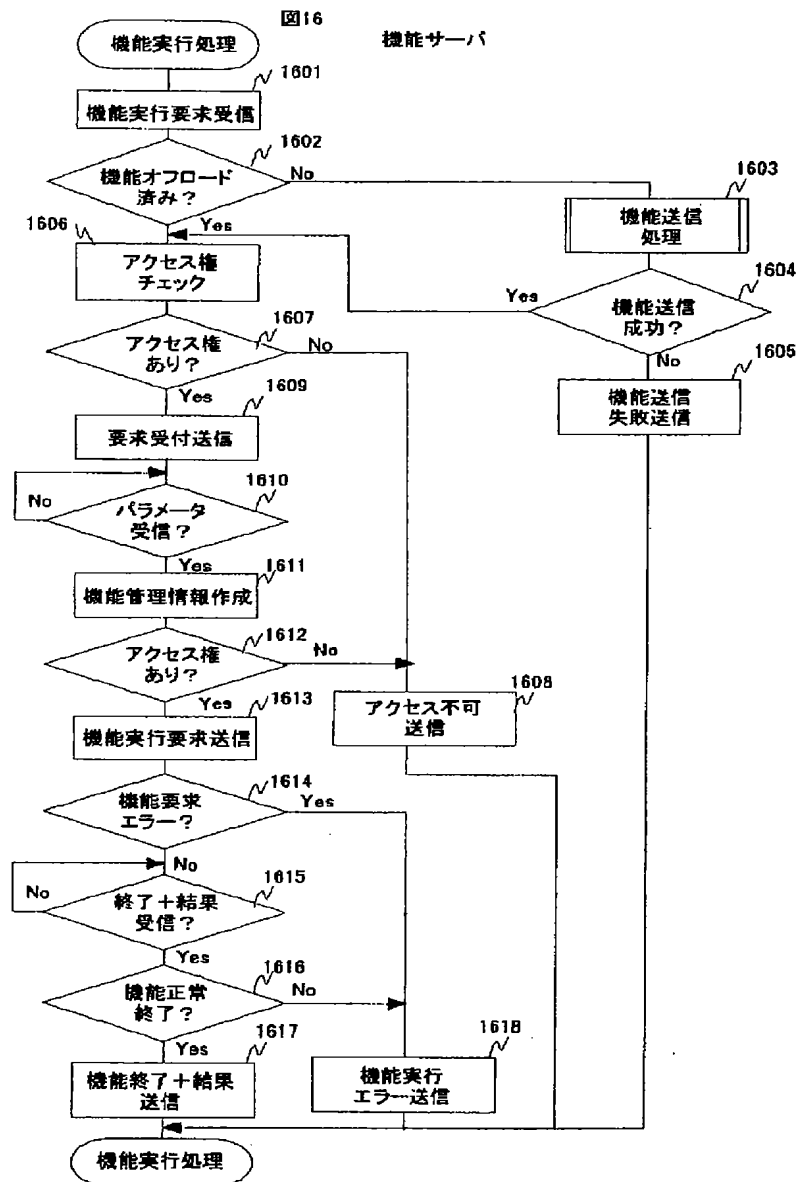
【図 14】



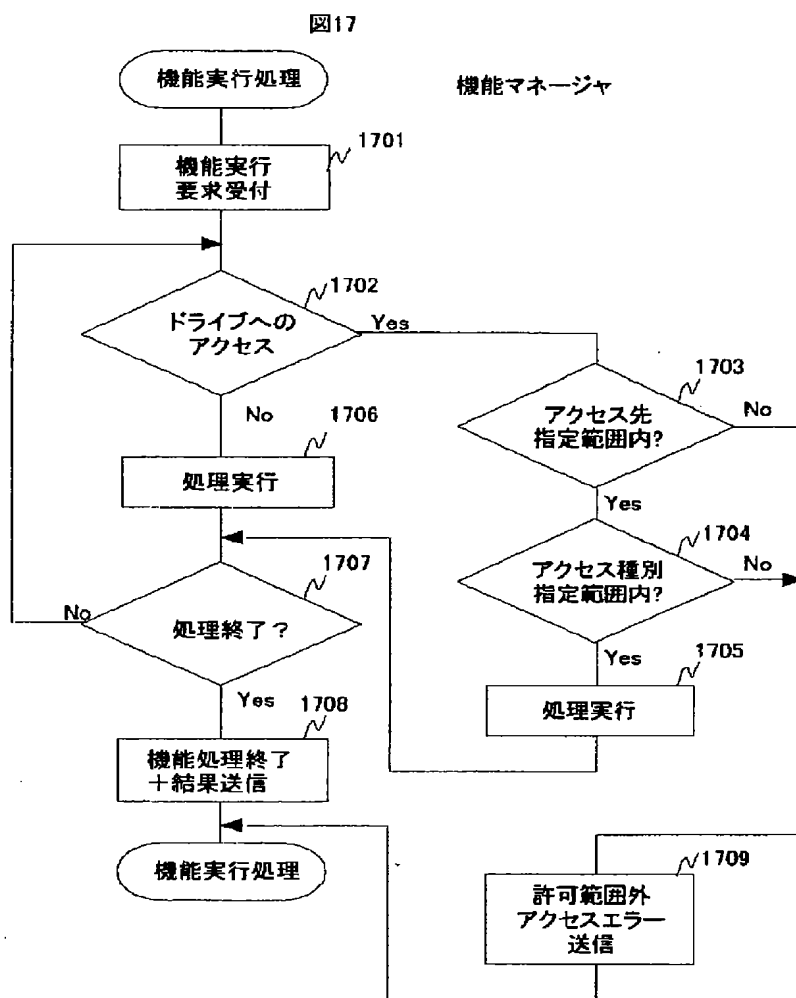
【図15】



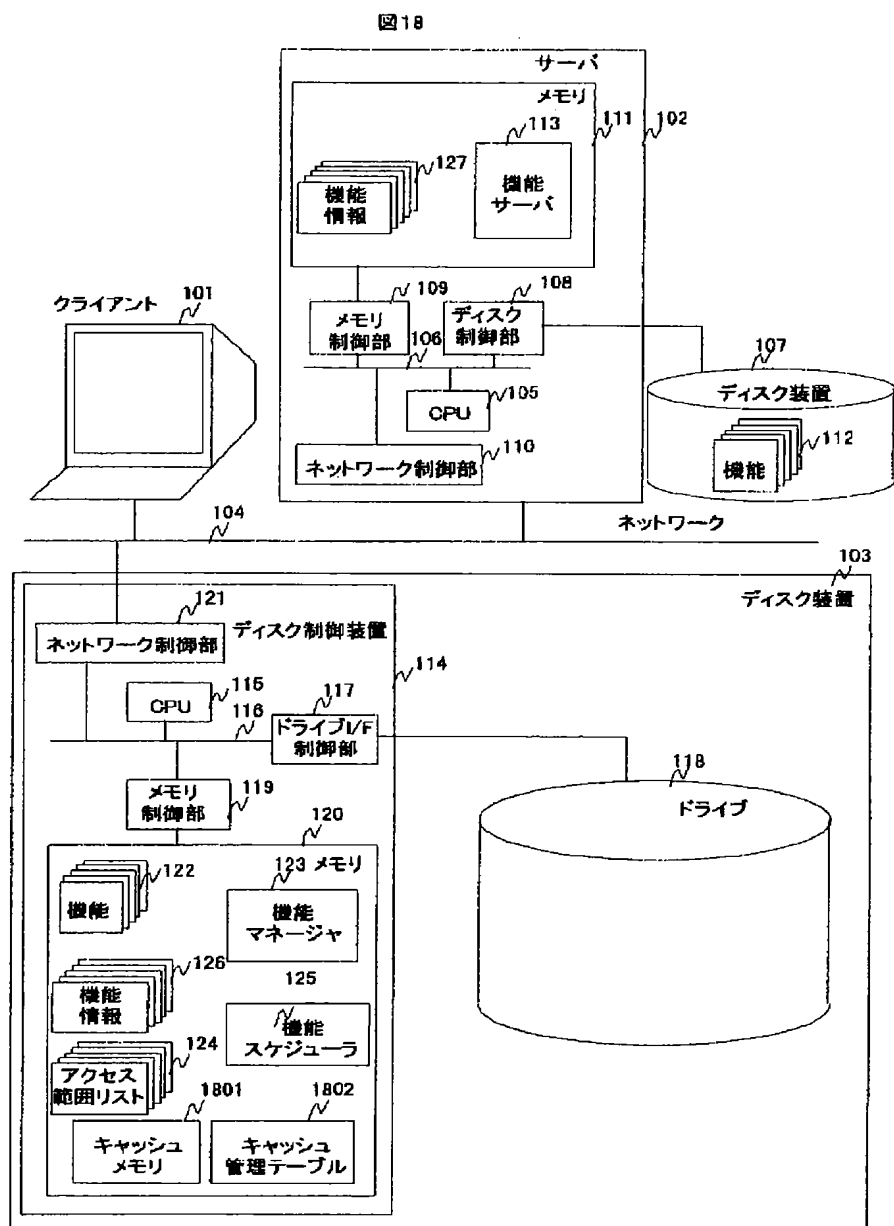
【図 16】



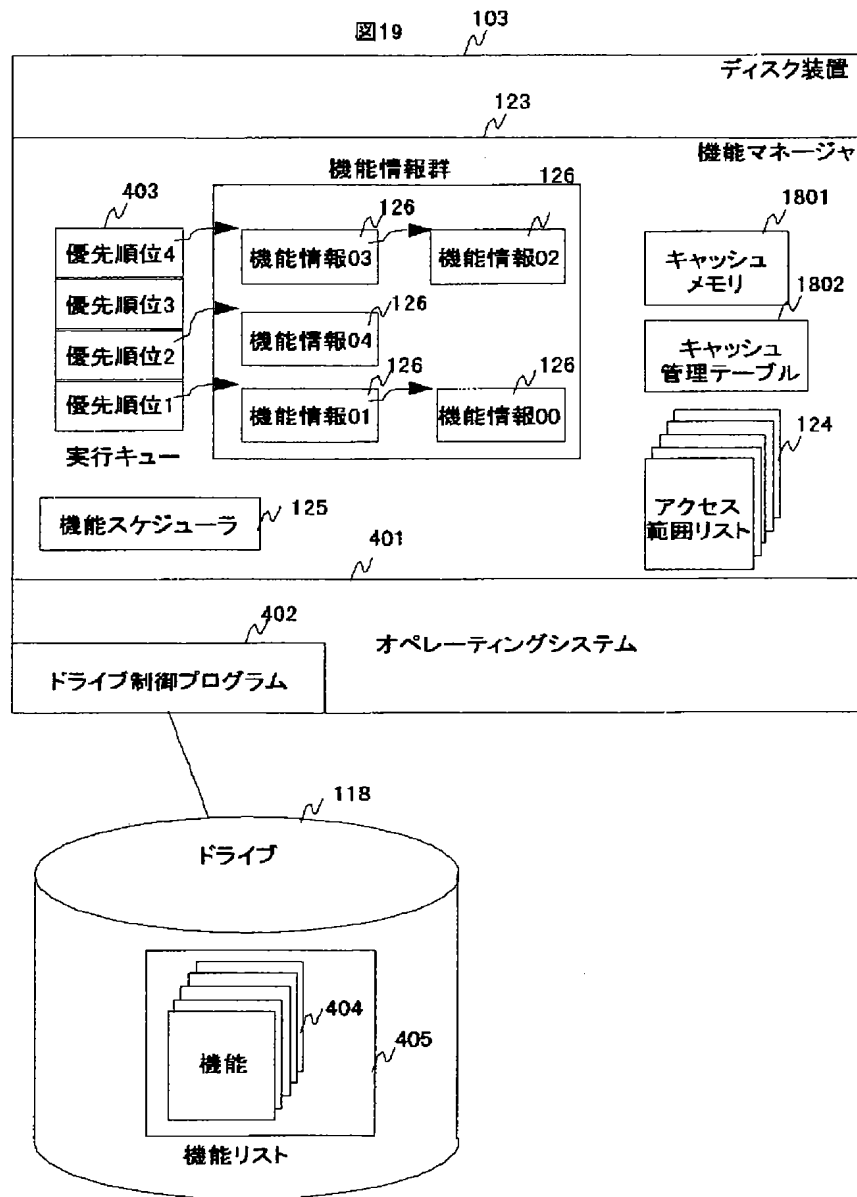
【図 17】



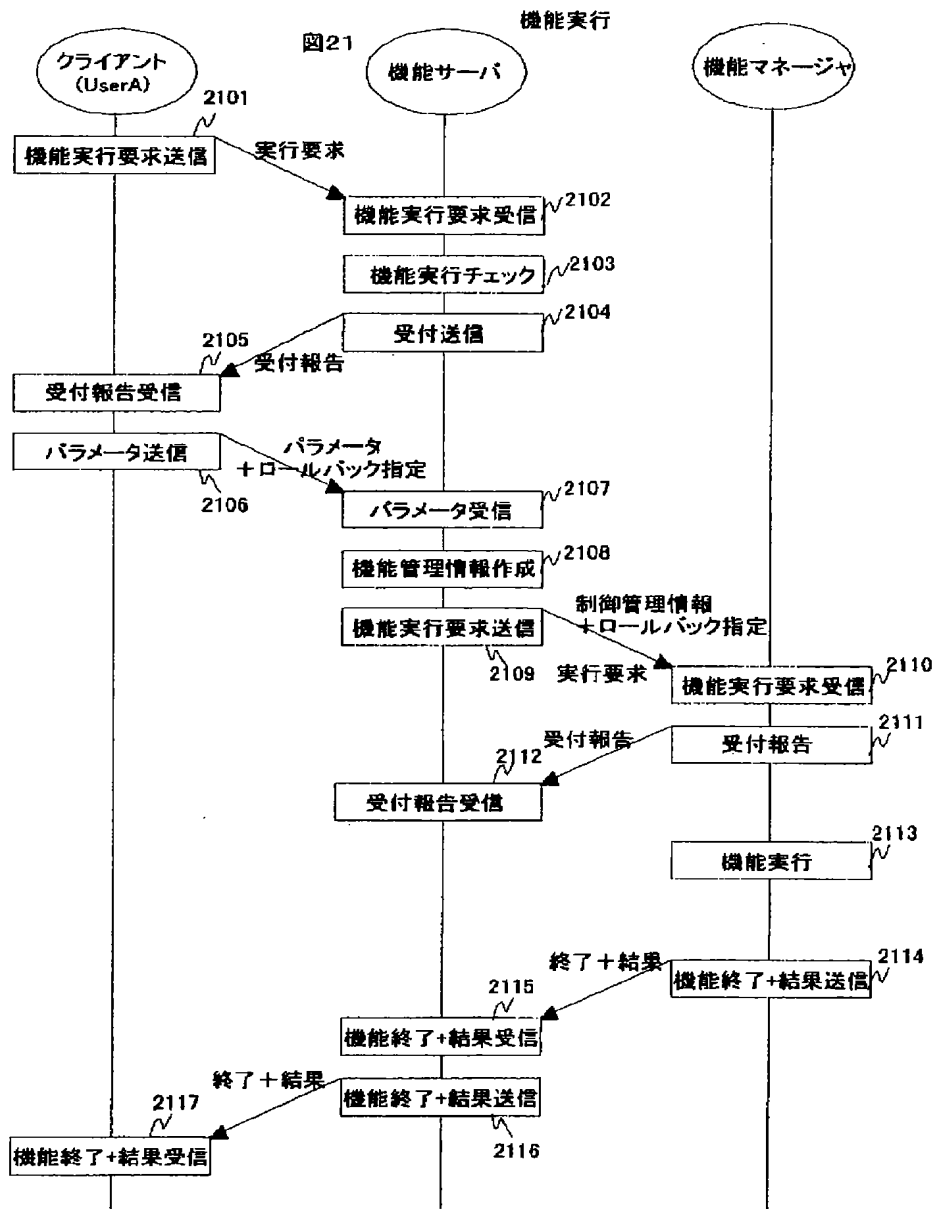
【図18】



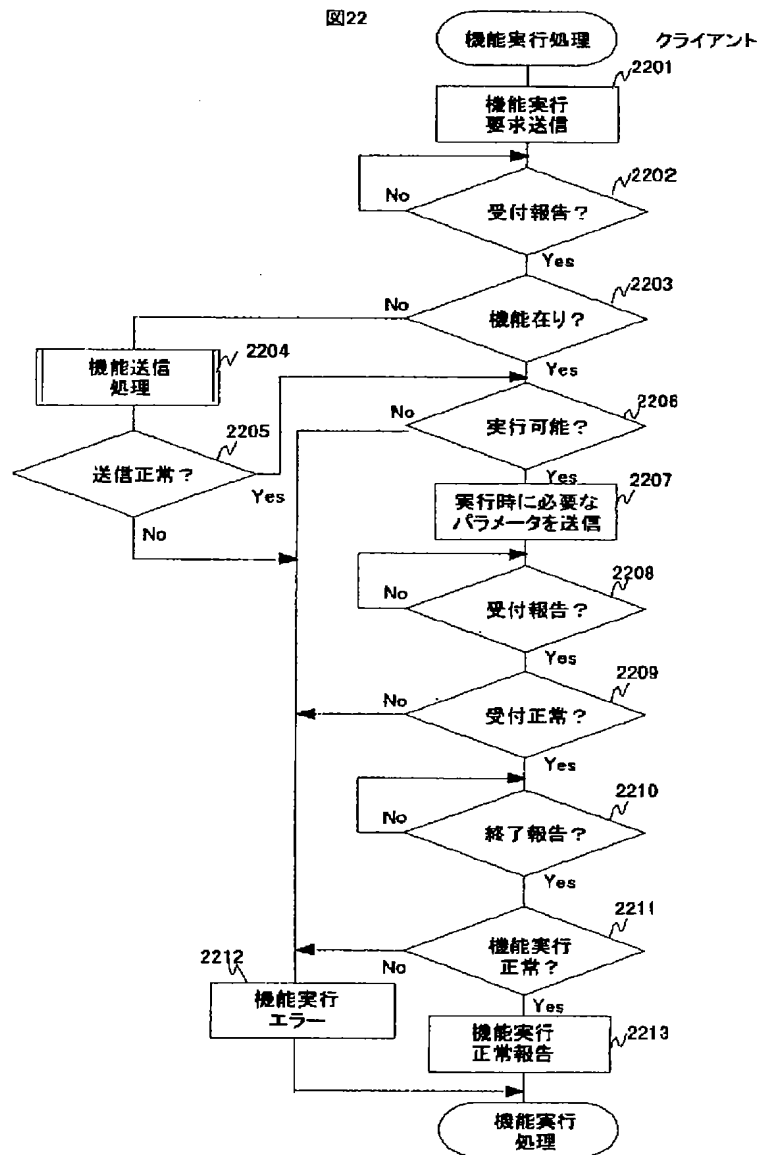
【図19】



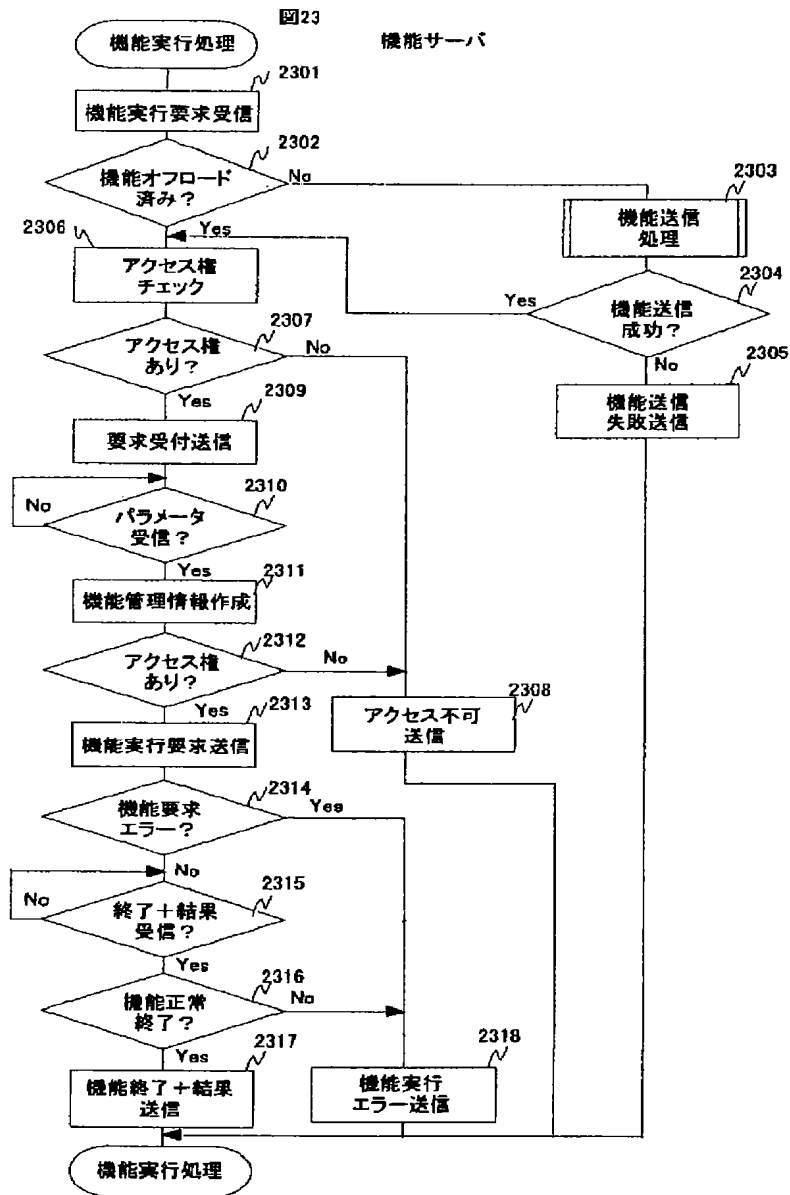
【図21】



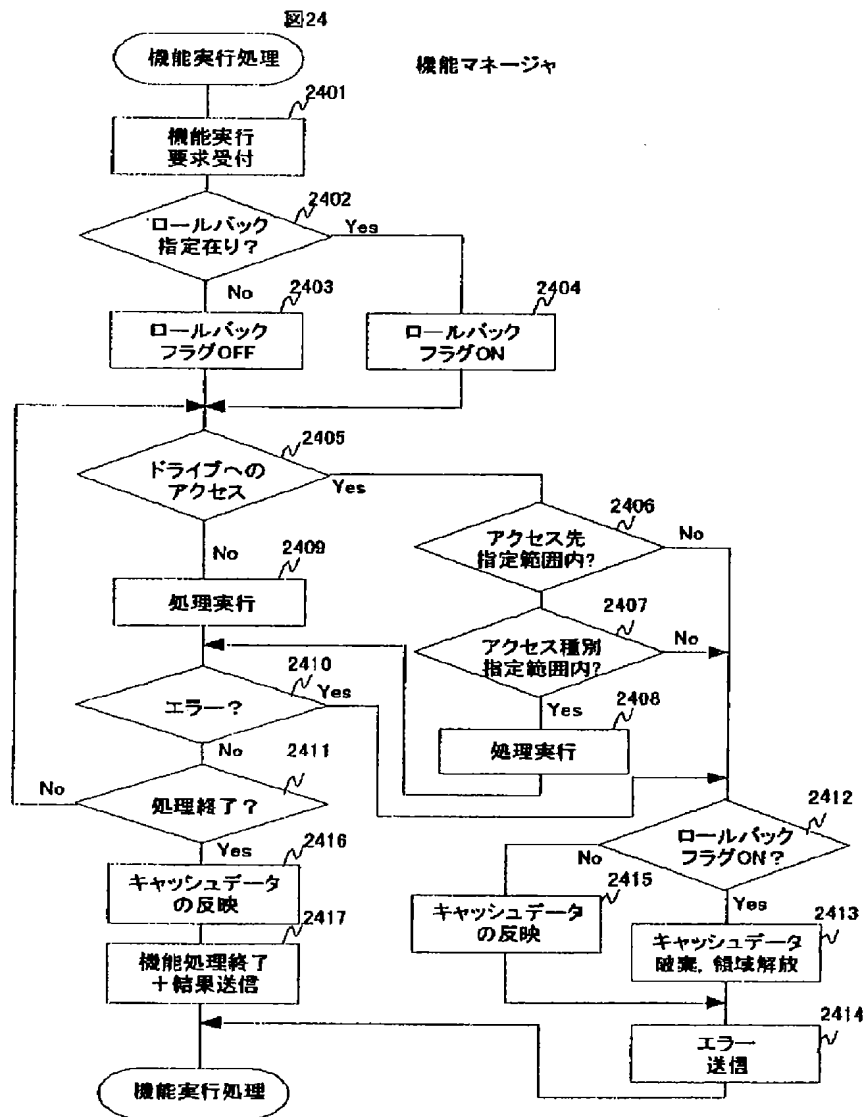
【図 22】



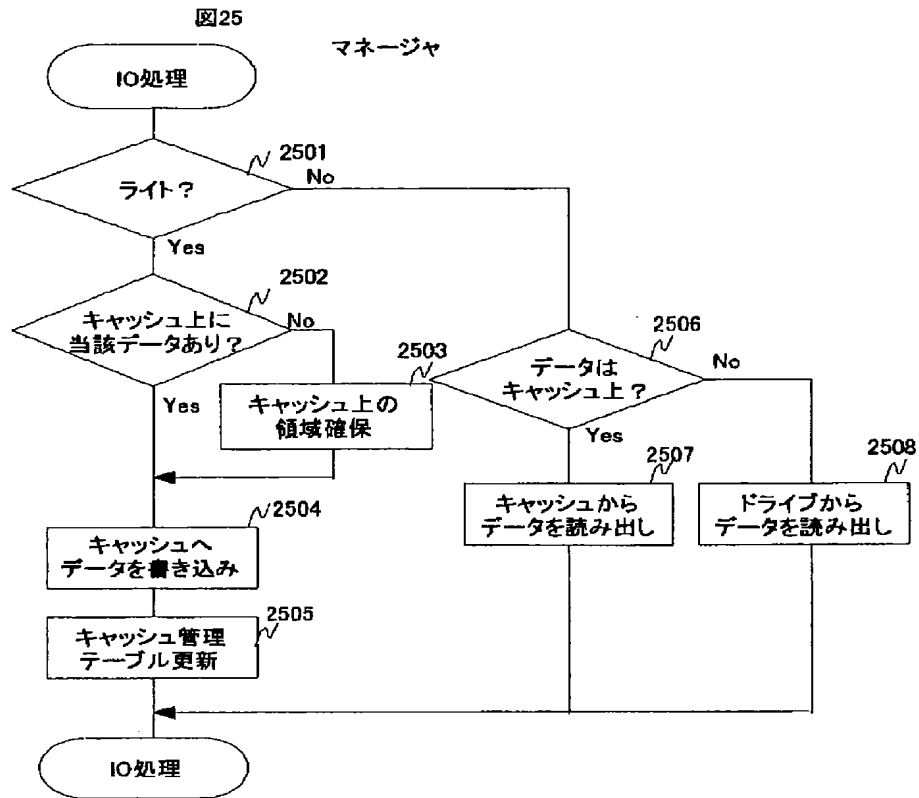
【図 23】



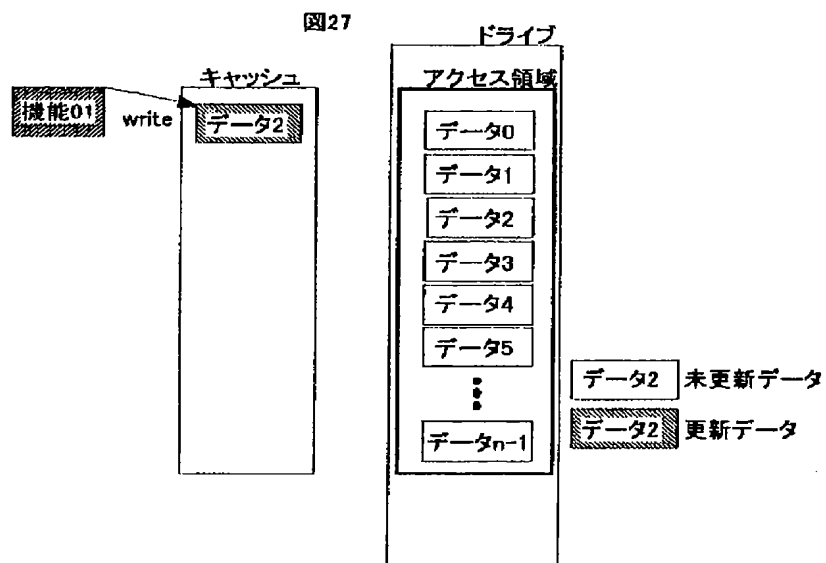
【図 24】



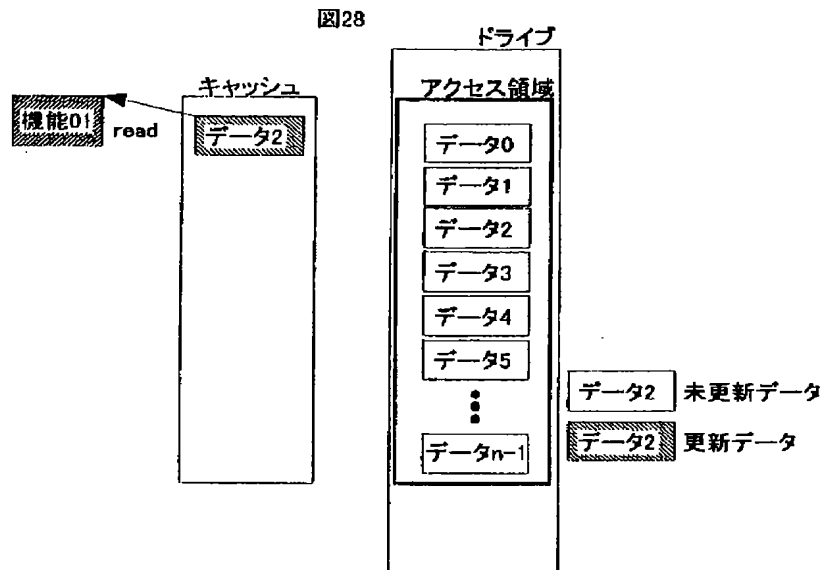
【図25】



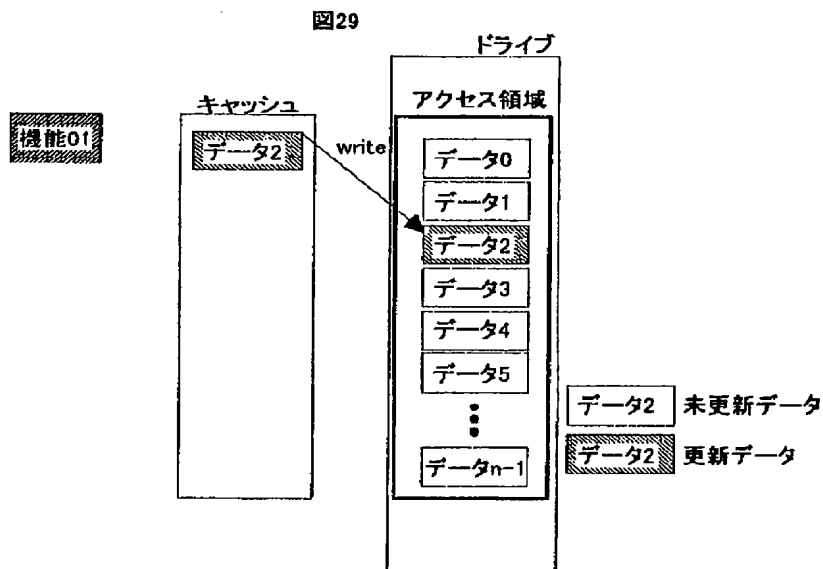
【図27】



【図 28】



【図 29】



フロントページの続き

(51) Int. Cl.⁷

G 0 6 F 12/14

識別記号

3 2 0

F I

G 0 6 F 12/14

ターコード (参考)

3 2 0 C

(72) 発明者 小田原 宏明

東京都国分寺市東恋ヶ窪一丁目280番地
株式会社日立製作所中央研究所内

F ターム(参考) 5B017 AA01 BA01 BA05 BA06 BB10
CA07 CA16
5B065 BA01 CA11 CC03 CE26 PA02
PA04 PA12 PA14
5B082 AA01 DC06 EA11 EA12 GA05
GA13 HA08